Dynamics of Belief: Abduction, Horn Knowledge Base And Database Updates

Radhakrishnan Delhibabu

Informatik 5, Knowledge-Based Systems Group
RWTH Aachen, Germany
delhibabu@kbsg.rwth-aachen.de

Abstract. The dynamics of belief and knowledge is one of the major components of any autonomous system that should be able to incorporate new pieces of information. In order to apply the rationality result of belief dynamics theory to various practical problems, it should be generalized in two respects: first it should allow a certain part of belief to be declared as immutable; and second, the belief state need not be deductively closed. Such a generalization of belief dynamics, referred to as base dynamics, is presented in this paper, along with the concept of a generalized revision algorithm for knowledge bases (Horn or Horn logic with stratified negation). We show that knowledge base dynamics has an interesting connection with kernel change via hitting set and abduction. In this paper, we show how techniques from disjunctive logic programming can be used for efficient (deductive) database updates. The key idea is to transform the given database together with the update request into a disjunctive (datalog) logic program and apply disjunctive techniques (such as minimal model reasoning) to solve the original update problem. The approach extends and integrates standard techniques for efficient query answering and integrity checking. The generation of a hitting set is carried out through a hyper tableaux calculus and magic set that is focused on the goal of minimality. The present paper provides a comparative study of view update algorithms in rational approach. For, understand the basic concepts with abduction, we provide an abductive framework for knowledge base dynamics. Finally, we demonstrate how belief base dynamics can provide an axiomatic characterization for insertion a view atom to the database. We give a quick overview of the main operators for belief change, in particular, belief update versus database update.

Keyword: AGM, Belief Revision, Belief Update, Horn Knowledge Base Dynamics, Kernel Change, Abduction, Hyber Tableaux, Magic Set, View update, Update Propagation.

1 Introduction

We live in a constantly changing world, and consequently our beliefs have to be revised whenever there is new information. When we encounter a new piece of information that contradicts our current beliefs, we revise our beliefs *rationally*.

In the last three decades, the field of computer science has grown substantially beyond mere number crunching, and aspires to imitate rational thinking of human beings. A separate branch of study, *artificial intelligence* (AI) has evolved, with a number of researchers attempting to represent and manipulate knowledge in a computer system. Much work has been devoted to study the statics of the knowledge, i.e. representing and deducting from fixed knowledge, resulting in the development of expert systems. The field of logic programming, conceived in last seventies, has proved to be an important tool for handling static knowledge. However, such fixed Horn knowledge based systems can not imitate human thinking, unless they are accomplish revising their knowledge in the light of new information. As mentioned before, this revision has to take place rationally. This has led to a completely new line of research, the **dynamics of belief**.

Studies in dynamics of belief are twofold: What does it mean to *rationally* revise a belief state? How can a belief state be represented in a computer and revised? The first question is more philosophical theory, and a lot of works have been carried out from epistemological perspective to formalize belief dynamics. The second question is computation oriented, and has been addressed differently from various perspectives of application. For example, a lot of algorithms have been proposed in logic programming to revise a Horn knowledge base or a database represented as a logic program; number of algorithms are there to carry out a view update in a rational database; algorithm to carry out diagnosis; algorithm for abduction reasoning and so on. We need the concept of ”change” in some form or other and thus need some axiomatic characterization to ensure that the algorithms are rational. Unfortunately, till this date, these two tracks remain separate, with minimal sharing of concepts and results. The primary purpose of the paper is to study these two developments and integrate them.

When a new piece of information is added to a Horn knowledge base (Delgrande 2008 and Delgrande & Peppas 2011), (Papini 2000) it may become inconsistent. Revision means modifying the Horn knowledge base in order to maintain consistency, by keeping the new information and removing the least possible previous information. In our case, update means revision and contraction, that is insertion and deletion in database perspective. Previous works (Aravindan & Dung 1994), (Aravindan 1995) have explained connections between contraction and knowledge base dynamics. Our Horn knowledge base dynamics is defined in two parts: an immutable part (Horn formulae) and updatable part (literals) (for definition and properties see the works of Nebel 1998, Segerberg 1998, Hansson et al 2001 and Fermé & Hansson 2001). Knowledge bases have a set of integrity constraints. In the case of finite knowledge bases, it is sometimes hard to see how the update relations should be modified to accomplish certain Horn knowledge base updates. .

2 Motivation

In the general case of arbitrary formulae, the revision problem for knowledge bases is hard to solve. So we restrict the revision problem to *Horn formulae.* The connection between belief change and database change is an interesting one since so far the two communities have independently considered two problems that are very similar, and our aim is to bring out this connection.

We aim to bridge the gap between philosophical and database theory. In such a case, Hansson's (Hansson 1997) kernel change is related to the abductive method. Aliseda's (Aliseda 2006) book on abductive reasoning is one of our key motivation. Wrobel's (Wrobel 1995) definition of first-order theory revision was helpful to frame our algorithm. On the other hand, we are dealing with the view update problem. Keller and Minker's (Keller 1985 and Minker 1996) work is one the motivation for the view update problem. In Figure 1 understand the concept of view update problem in rational way. Figure show that foundation form Belief Revision theory, intermediate step handle to Horn knowledge base, this step very impairment that agent have background knowledge and he/she made decision with postulate may require to process next step. Target of the application is connect database updates via Horn knowledge base with abduction reasoning. All clear procedure shown in each section.

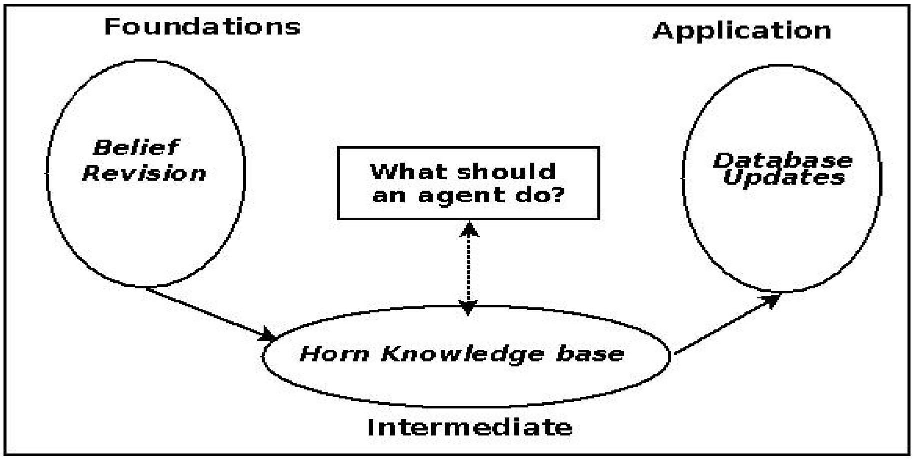


Fig. 1. Layout of the paper

Following example illustrates the motivation of the paper:

Example 1. Consider a database with an (immutable) rule that a staff member is a person who is currently working in a research group under a chair. Additional (updatable) facts are that matthias and gerhard are group chairs, and delhibabu and aravindan are staff members in group infor1. Our first integrity constraint (IC) is that each research group has only one chair ie. $\forall x, y, z$ (y=z) ← group_chair(x,y) $\wedge$ group_chair(x,z). Second integrity constraint is that a person can be a chair for only one research group ie. $\forall x, y, z$ (y=z)← group_chair(y,x) $\wedge$ group_chair(z,x).

Immutable part: staff_chair(x,y)← staff_group(x,z),group_chair(z,y).

Updatable part: group_chair(infor1,matthias)←
group_chair(infor2,gerhard)←
staff_group(delhibabu,infor1)←
staff_group(aravindan,infor1)←

Suppose we want to update this database with the information, staff_chair(aravindan,gerhard); From the immutable part, we can deduce that this can be achieved by asserting staff_group(aravindan,z) $\bigwedge$ group_chair(z,gerhard)

If we are restricted to definite clauses, there are three plausible ways to do this: first case is, aravindan and gerhard belong to infor1, i.e, staff_group(aravindan,infor1) $\bigwedge$ group_chair(info1,gerhard). We need to delete both base facts group_chair(infor1,matthias)← and group_chair(infor2,gerhard)←, because our first IC as well as second IC would be violated otherwise. In order to change the view, we need to insert group_chair(infor1,gerhard)← as a base fact. Assume that we have an algorithm that deletes the base facts staff_group(delhibabu,infor1)← from the database. But, no rational person will agree with such an algorithm, because the fact staff_group(delhibabu,infor1)← is not ”relevant” to the view atom.

Second case, aravindan and gerhard belong to infor2, that is staff_group(aravindan,infor2) $\bigwedge$ group_chair(infor2,gerhard). Simply, insert the new fact staff_group(aravindan,infor2)← to change the view. Suppose an algorithm deletes the base facts staff_group(aravindan,infor1)← from the database, then it can not be ”rational” since these facts are not ”relevant” to the view atom.

Third case, aravindan and gerhard belong to infor3 (free assignment of the group value), that is staff_group(aravindan,infor3) $\bigwedge$ group_chair(info3,gerhard). Suppose, we insert new base fact group_chair(infor3,gerhard) ←, our second IC does not follow. Suppose an algorithm inserts the new base fact staff_group(aravindan,infor2)← or staff_group(aravindan,infor1)← is deleted, then it can not be ”rational”.

The above example highlights the need for some kind of ”relevance policy” to be adopted when a view atom is to be inserted to a deductive database. How many such axioms and policies do we need to characterize a ”good” view update? When are we sure that our algorithm for view update is ”rational”? Clearly, there is a need for an axiomatic characterization of view updates. By axiomatic characterization, we mean explicitly listing all the rationality axioms that are to be satisfied by any algorithm for view update.

The basic idea in (Behrend & Manthey 2008), (Aravindan & Baumgartner 1997) is to employ the model generation property of hyper tableaux and magic set to generate models, and read off diagnosis from them. One specific feature of this diagnosis algorithm is the use of semantics (by transforming the system description and the observation using an initial model of the correctly working system) in guiding the search for a diagnosis. This semantical guidance by program transformation turns out to be useful for database updates as well. More specifically we use a (least) Herbrand model of the given database to transform it along with the update request into a logic program in such a way that the models of this transformed program stand for possible updates.

We discuss two ways of transforming the given database together with the view update (insert and delete) request into a logic program resulting in two variants of view update algorithms. In the first variant, a simple and straightforward transformation is employed. Unfortunately, not all models of the transformed program represent a rational update using this approach. The second variant of the algorithm uses the least Herbrand model of the given database for the transformation. In fact what we referred to as offline preprocessing before is exactly this computation of the least Herbrand model. This variant is very meaningful in applications where views are materialized for efficient query answering. The advantage of using the least Herbrand model for the transformation is that all models of the transformed logic program (not just the minimal ones) stand for a rational update.

When dealing with the revision of a Horn knowledge base (both insertions and deletions), there are other ways to change a Horn knowledge base and it has to be performed automatically also (Fermé 1992 and Rodrigues& Benevidas 1994). Considering the information, change is precious and must be preserved as much as possible. The *principle of minimal change* (Gärdenfors 1998, Dalal 1988 and Herzig & Rifi 1999), (Schulte 1999) can provide a reasonable strategy. On the other hand, practical implementations have to handle contradictory, uncertain, or imprecise information, so several problems can arise: how to define efficient change in the style of Carlos Alchourr*ó*n, Peter Gärdenfors, and David Makinson (AGM) (Alchourron et al. 1985b); what result has to be chosen (Lakemeyer 1995), (Lobo & Trajcevski 1997), (Nayak et al. 2006); and finally, according to a practical point of view, what computational model to explore for the Horn knowledge base revision has to be provided?

The rest of the paper is organized as follows: First we start with preliminaries in Section 3. In Section 4, we give a quick overview of belief changes and belief update. In Section 5, we introduce knowledge base dynamics along with the concept of generalized revision, and revision operator for knowledge base. Section 6 studies the relationship between knowledge base dynamics and abduction and shows how abductive procedures could be used to realize revision. In Section 7, we give a quick overview of belief update versus knowledge base update. In Section 8, we discuss an important application of knowledge base dynamics in providing an axiomatic characterization for insertion view atoms to databases; and nature of view update problem for incomplete to complete information shown. We give a quick overview of belief update versus database update in Section 9. In Section 10, we provide an abductive framework for Horn knowledge base dynamics in first order version. In Section 11, we give brief overview of related works. In Section 12, we draw conclusions with a summary of our contribution and indicate future directions of our investigation.

3 Preliminaries

We consider a propositional language $\mathcal{L}_{\mathcal{P}}$ defined from a finite set of propositional variables $\mathcal{P}$ and the standard connectives. We use $a, b, x, y, ...\varphi, \phi, \psi, ...$ for propositional formulae. Sets of formulae are denoted by upper case Roman letters $A, B, F, K, ...$ A literal is an atom (positive literal), or a negation of an atom (negative literal).

For any formula φ, we write $E(\varphi)$ to mean the set of the elementary letters that occur in φ. The same notation also applies to a set of formulae. For any set F of formulae, $L(F)$ represents the sub-language generated by $E(F)$, i.e. the set of all formulae φ with $E(\varphi) \subseteq E(F)$.

Horn formulae are defined (Delgrande & Peppas 2011) as follows:

1. Every $a \in \varPhi$ where $\varPhi \in \mathcal{L}_{\mathcal{P}} \cup \{\bot\}$, a and $\neg a$ are Horn clauses.
2. $a \leftarrow a_1 \wedge a_2 \wedge ... \wedge a_n$ is a Horn clause, where $n \geq 0$ and $a, a_i \in \varPhi$ $(1 \leq i \leq n)$.
3. Every Horn clause is a Horn formula, a is called head and a_i is body of the Horn formula.
4. If φ and ψ are Horn formulae, so is $\varphi \wedge \psi$.

A definite Horn clause is a finite set of literals (atoms) that contains exactly one positive literal which is called the head of the clause. The set of negative literals of this definite Horn clause is called the body of the clause. A Horn clause is non-recursive, if the head literal does not occur in its body. We usually denote a Horn clause as head←body. Let $\mathcal{L}_{\mathcal{H}}$ be the set of all Horn formulae with respect to $\mathcal{L}_{\mathcal{P}}$. A formula ϕ is a syntactic consequence within $\mathcal{L}_{\mathcal{P}}$ of a set $\varGamma$ of formulas if there is a formal proof in $\mathcal{L}_{\mathcal{P}}$ of ϕ from the set $\varGamma \vdash_{\mathcal{L}_{\mathcal{P}}} \phi$.

A Horn logic with stratified negation (Jackson & Schulte 2008) is similar to a set of Horn formulae. An immutable part is a function-free clause of the form $a \leftarrow a_1 \wedge a_2 \wedge ... \wedge a_n$, with $n \geq 1$ where a is an atom denoting the immutable part's head and $a_1 \wedge a_2 \wedge ... \wedge a_n$ are literals. i.e. positive or negative atoms, representing the body of the Horn clauses.

Formally, a finite Horn knowledge base KB (Horn or Horn logic with stratified negation) is defined as a finite set of formulae from language $\mathcal{L}_{\mathcal{H}}$, and divided into three parts: an immutable theory KB_I is a Horn formula (head←body), which is the fixed part of the knowledge; updatable theory KB_U is a Horn clause (head←); and integrity constraint KB_{IC} representing a set of clauses (Horn logic with stratified negation) (←body).

Definition 1 (Horn Knowledge Base). *A Horn knowledge base, KB is a finite set of Horn formulae from language $\mathcal{L}_{\mathcal{H}}$, s.t $KB = KB_I \cup KB_U \cup KB_{IC}$, $KB_I \cap KB_U = \varnothing$ and $KB_U \cap KB_{IC} = \varnothing$.*

Horn knowledge base change deals with situations in which an agent has to modify its beliefs about the world, usually due to new or previously unknown incoming information, also represented as formulae of the language. Common operations of interest in Horn knowledge base change are the expansion of an agent's current Horn knowledge base KB by a given Horn clause φ (usually denoted as KB+φ), where the basic idea is to add regardless of the consequences, and the revision of its current beliefs by φ (denoted as KB * φ), where the intuition is to incorporate φ into the current beliefs in some way while ensuring consistency of the resulting theory at the same time. Perhaps the most basic operation in Horn knowledge base change, like belief change, is that of contraction AGM (Alchourron et al. 1985b), which is intended to represent situations in which an agent has to give up φ from its current stock of beliefs (denoted as KB-φ).

Definition 2 (AGM Contraction). *Let KB be a Horn knowledge base, and α a belief that is present in KB. Then* contraction *of KB by α, denoted as $KB - \alpha$, is a consistent belief set that ignore α*

Definition 3 (Levi Identity). *Let - be an AGM contraction operator for KB. A way to define a revision is by using Generalized Levi Identity:*

$$KB * \alpha \ = \ (KB - \neg\alpha) \cup \alpha$$

Then, the revision can be trivially achieved by expansion, and the axiomatic characterization could be straightforwardly obtained from the corresponding characterizations of the traditional models (Alchourron, CE et al 1985). The aim of our work is not to define revision from contraction, but rather to construct and axiomatically characterize revision operators in a direct way.

4 Belief Changes

Working at an abstract philosophical level, the aim of belief dynamics is to formalize the rationality of change, without worrying much about the syntactic representation of belief. However, it is not possible to completely ignore belief representation, and works on belief dynamics assume as little necessary things as possible about the representation of the belief. In this Section based on Konieczny's (Konieczny 2011) work, we recall the definition of the main belief change operators and the links between them. We focus on the classical case, where the belief represent use propositional logic. This is a very quick presentation of belief change theory. For a complete introduction the reader is referred to seminal books on belief revision ((Gärdenfors 1992 & 1998), (Hansson 1997a), (Rott 2001)) or the recent special issue of Journal of Philosophical Logic on the 25 Years of AGM Theory (Ferme & Hansson 2011).

A belief base K is a finite set of propositional formulae. In order to simplify the notations we identify the base K with the formula. which is the conjunction of the formulae of K [1]

[1] There are two major interpretations of belief bases. One of them, supported by Dalal 1998, uses belief bases as mere expressive devices; hence if Cn(B1) = Cn(B2) then

Belief revision aims at changing the status of some beliefs in the base that are contradicted by a more reliable piece of information. Several principles are govern this revision operation:

- First is the primacy of update principle: the new piece of information has to be accepted in the belief base after the revision. This is due to the hypothesis that the new piece of information is more reliable than the current beliefs [2]
- Second is the principle of coherence: the new belief base after the revision should be a consistent belief base. Asking the beliefs to be consistent is a natural requirement if one wants to conduct reasoning tasks from her belief base
- Third is the principle of minimal change: the new belief base after the revision should be as close as possible from the current belief base. This important principle aims at ensuring that no unnecessary information (noise) is added to the beliefs during the revision process, and that no unnecessary information is lost during the process: information/beliefs are usually costly to obtain, we do not want to throw them away without any serious reason.

let ψ and μ be two formulae denoting respectively the belief base, and a new piece of information. Then $\psi \circ \mu$ is a formula representing the new belief base. An operator $\circ$ is an AGM belief revision operator if it satisfies the following properties.

Definition 4 (Belief revision).

(R1) $\psi \circ \mu$ *implies* μ.
(R2) If $\psi \wedge \mu$ *is satisfiable, then* $\psi \circ \mu \equiv \psi \wedge \mu$.
(R3) If μ *is satisfiable, then so is* $\psi \circ \mu$.
(R4) If $\psi_1 \equiv \psi_2$ *and* $\mu_1 \equiv \mu_2$, *then* $\psi_1 \circ \mu_1 \equiv \psi_2 \circ \mu_2$.
(R5) $(\psi \circ \mu) \wedge \phi$ *implies* $\psi \circ (\mu \wedge \phi)$.
(R6) If $(\psi \circ \mu) \wedge \phi$ *is satisfiable, then* $\psi \circ (\mu \wedge \phi)$ *implies* $(\psi \circ \mu) \wedge \phi$.

When one works with a finite propositional language the above postulates, proposed by Katsuno and Mendelzon (Katsuno, H & Mendelzon, AO. 1991b), are equivalent to AGM ((Alchourron et al 1985b) and (Gärdenfors 1998)) (R1) states that the new piece of information must be believed after the revision. (R2) says that when there is no conflict between the new piece of information and the current belief, the revision is just the conjunction. (R3) says that revision always a consistent belief base, unless the new piece of information is not consistent. (R4) is an irrelevance of syntax condition, it states that logically equivalent bases

B1 and B2 represent the same belief state and yield the same outcome under all operations of change. The other, more common approach treats inclusion in the belief base as epistemically significant. The belief base contains those sentences that have an epistemic standing of their own (Ferme 2011)

[2] If this is not the case one should use a non-prioritized revision operator (Hansson 1997b)

must lead to the same result. (R5) and (R6) give conditions on the revision by a conjunction.

AGM also defined contraction operators, that aim to remove some piece of information from the beliefs of the agent. These contraction operators are closely related to revision operators, since each contraction operator can be used to define a revision operator, through the Levy identity and conversely each revision operator can be used to define a contraction operator through the Harper identity ((Alchourron et al 1985b) and (Gärdenfors 1998)). So one can study indifferently revision or contraction operators. So we focus on revision here.

Several representation theorems, that give constructive ways to define AGM revision/ contraction operators, have been proposed, such as partial meet contraction/revision (Alchourron et al 1985b), epistemic entrenchments (Gärdenfors 1992) and (Gärdenfors & Makinson 1988), safe contraction (Alchourron et al 1985a), etc. In (Katsuno & Mendelzon 1991b and 1992) (Benferhat et al. 2005), Katsuno and Mendelzon give a representation theorem, showing that each revision operator corresponds to a faithful assignment, that associates to each base a plausibility preorder on interpretations (this idea can be traced back to Grove systems of spheres (Grove 1988)).

Assume a total pre-order $\leq_\psi$ on W (set of possible world). That is to say, KB $= min(W, \leq_\psi)$. As usual we take $\leq_\psi$ to be an ordering of plausibility on the worlds, with worlds lower down in the ordering seen as more plausible. In what follows $\simeq_\psi$ will always denote the symmetric closure of $\leq_\psi$, i.e., $W1 \simeq_\psi W2$ iff both $W1 \leq_\psi W2$ and $W2 \leq_\psi W1$.

Definition 5 ((Konieczny 2011)). *A faithful assignment is a function mapping each base ψ to a pre-order $\leq_\psi$ over interpretations such that*

1. *if $\omega \models \psi$ and $\omega' \models \psi$, then $\omega \simeq_\psi \omega'$*
2. *if $\omega \models \psi$ and $\omega' \not\models \psi$, then $\omega <_\psi \omega'$*
3. *if $\psi \equiv \psi'$, then $\leq_\psi = \leq_{\psi'}$*

Theorem 1 ((Katsuno & Mendelzon 1991b and 1991b)). *An operator $\circ$ is an AGM revision operator (i.e. it satisfies (R1)-(R6)) if and only if there exists a faithful assignment that maps each base ψ to a total pre-order $\leq_\psi$ such that $mod(\psi \circ \mu) = min(mod(\mu), \leq_\psi)$.*

Proof. Follows from the definition 5 and the result of Konieczny 2011.

One of the main problems of this characterization of belief revision is that it does not constrain the operators enough for ensuring a good behavior when we do iteratively several revisions. So one needs to add more postulates and to represent the beliefs of the agent with a more complex structure than a simple belief base. In (Darwiche & Pearl 1997) Darwiche and Pearl proposed a convincing extension of AGM revision. This proposal have improved by an additional condition in ((Booth & Meyer 2006) and (Jin & Thielscher 2007), (Konieczny, et al. 2010) and (Konieczny & Pino Prez 2008)) define improvement operators that are a generalization of iterated revision operators.

Whereas belief revision should be used to improve the beliefs by incorporating more reliable pieces of evidence, belief update operators aim at maintaining the belief base up-to-date, by allowing modifications of the base according to a reported change in the world. This distinction between revision and update was made clear in Katsuno & Mendelzon 1991 and 1992, where Katsuno & Mendelzon 1991 proposed postulates for belief update.

Definition 6 (Belief Update).

An operator $\diamond$ is a (partial) update operator if it satisfied the properties (U1)-(U8). It is a total update operator if it satisfies the property (U1)-(U5),(U8),(U9).

(U1) $\psi \diamond \mu$ implies μ.
(U2) ifψ implies μ, then $\psi \diamond \mu \equiv \psi$
(U3) ifψ not implies $\bot$ and μ not implies $\bot$ then $\psi \diamond \mu$ not implies $\bot$
(U4) If $\psi_1 \equiv \psi_2$ and $\mu_1 \equiv \mu_2$, then $\psi_1 \diamond \mu_1 \equiv \psi_2 \diamond \mu_2$.
(U5) If $(\psi \diamond \mu) \wedge \phi$ implies $\psi \diamond (\mu \wedge \phi)$
(U6) If $(\psi \diamond \mu_1)$ implies μ_2 and $(\psi \diamond \mu_2)$ implies μ_2 then $\psi \diamond \mu_1 \equiv \psi \diamond \mu_2$
(U7) If ψ is a complete formula, then $(\psi \diamond \mu_1) \wedge (\psi \diamond \mu_2)$ implies $\psi \diamond (\mu_1 \vee \mu_2)$
(U8) $(\psi_1 \vee \psi_2) \diamond \mu \equiv (\psi_1 \diamond \mu) \vee (\psi_2 \diamond \mu)$
(U9) If ψ is a complete formula and $(\psi \diamond \mu) \wedge \psi$ not implies $\bot$ then $\psi \diamond (\mu \wedge \psi)$ implies $(\psi \diamond \mu) \wedge \psi$

Most of these postulates are close to the ones of revision. The main differences lie in postulate (U2) that is much weaker than (R2): conversely to revision, even if the new piece of information is consistent with the belief base, the result is generally not simply the conjunction. This illustrates the fact that revision can be seen as a selection process of the most plausible worlds of the current beliefs with respect to the new piece information, whereas update is a transition process: each world of the current beliefs have to be translated to the closest world allowed by the new piece of information. This world-by-world treatment is expressed by postulate (U8).

As for revision, there is a representation theorem in terms of faithful assignment.

Definition 7 ([25]). *A faithful assignment is a function mapping each interpretation ω to a pre-order $\leq_\omega$ over interpretations such that if $\omega \neq \omega'$, then $\omega <_\omega \omega'$*

Theorem 2. *An update operator $\diamond$ satisfies (U1)-(U8) if and only if there exists a faithful assignment that maps each interpretation ω to a partial pre-order $\leq_\omega$ such that $mod(\psi \diamond \mu) = \bigcup_{\omega \models \psi} min(mod(\mu), \leq_\omega)$.*

Proof. Follows from the observation and the result of Konieczny 2011.

But there is also a second theorem corresponding to total pre-orders.

Theorem 3. *An update operator $\diamond$ satisfies (U1)-(U5), (U8) and (U9) if and only if there exists a faithful assignment that maps each interpretation ω to a total preorder $\leq_\omega$ such that $mod(\psi \diamond \mu) = \bigcup_{\omega \models \psi} min(mod(\mu), \leq_\omega)$.*

Proof. Follows from the observation and the result of Konieczny 2011.

Definition 8 ([3]). *Let M=(W.w) be a K-model and μ a formula. A k-model $M' = (W', w')$ is called a possible resulting k-model after updating M with μ if and only if the following conditions hold:*

1. *M'$\models \mu$;*
2. *there does not exist another k-model $M'' = (W'', w'')$ such that $M'' \models \mu$ and $M'' <_M M'$.*

The set of all possible resulting k-models after updating M with μ as Res(M,μ).

Theorem 4. *Knowledge update operator $\diamond$ defined in definition 8 satisfies (U1)-(U9).*

Proof. Follows from the definition 8 and the result of Baral & Zhang 2005.

Note 1. Horn knowledge base is a subset of belief base, $KB \subseteq B$, so everything that follows for belief base, also follows for Horn Knowledge base.

5 Horn Knowledge base dynamics

In the AGM framework, a belief set is represented by a deductively closed set of propositional formulae. While such sets are infinite, they can't always be finitely representable. However, working with deductively closed, infinite belief sets is not very attractive from a computational point of view. The AGM approach to belief dynamics is very attractive in its capture the rationality of change, but it is not always easy to implement either Horn formula based partial meet revision or generalized kernel revision. In artificial intelligence and database applications, what is required is to represent the knowledge using a finite Horn knowledge base. Further, a certain part of the knowledge is treated as immutable and should not be changed.

AGM (Alchourron et al. 1985b) proposed a formal framework in which revision is interpreted as belief change. In this section, we focus on the Horn knowledge base revision and propose new rationality postulates that are adopted from AGM postulates for revision.

Definition 9. *Let KB be a Horn knowledge base with an immutable part KB_I. Let α and β be any two Horn clauses from $\mathcal{L_H}$. Then, α and β are said to be* KB-equivalent *iff the following condition is satisfied: $\forall$ set of Horn clauses E $\subseteq \mathcal{L_H}$ and IC: $KB_I \cup E \cup IC \vdash \alpha$ iff $KB_I \cup E \cup IC \vdash \beta$.*

These postulates stem from three main principles: the new item of information has to appear in the revised Horn knowledge base, the revised base has to be consistent and revision operation has to change the least possible beliefs. Now we consider the revision of a Horn clause α with respect to KB, written as $KB * \alpha$. The rationality postulates for revising α from KB can be formulated.

Definition 10 (Rationality postulates for Horn knowledge base revision).

*(KB*1)* Closure: *$KB * \alpha$ is a Horn knowledge base.*
*(KB*2)* Weak Success: *if α is consistent with $KB_I \cup KB_{IC}$ then $\alpha \subseteq KB * \alpha$.*
*(KB*3.1)* Inclusion: $KB * \alpha \subseteq Cn(KB \cup \alpha)$.
*(KB*3.2)* Immutable-inclusion: $KB_I \subseteq Cn(KB * \alpha)$.
*(KB*4.1)* Vacuity 1: *if α is inconsistent with $KB_I \cup KB_{IC}$ then $KB * \alpha = KB$.*
*(KB*4.2)* Vacuity 2: *if $KB \cup \alpha \nvdash \bot$ then $KB * \alpha = KB \cup \alpha$.*
*(KB*5)* Consistency: *if α is consistent with $KB_I \cup KB_{IC}$ then $KB * \alpha$ consistent with $KB_I \cup KB_{IC}$.*
*(KB*6)* Preservation: *If α and β are KB-equivalent, then $KB * \alpha \leftrightarrow KB * \beta$.*
*(KB*7.1)* Strong relevance: $KB * \alpha \vdash \alpha$ *if* $KB_I \nvdash \neg\alpha$
*(KB*7.2)* Relevance: *If $\beta \in KB \backslash KB * \alpha$, then there is a set KB' such that $KB * \alpha \subseteq KB' \subseteq KB \cup \alpha$, KB' is consistent $KB_I \cup KB_{IC}$ with α, but $KB' \cup \{\beta\}$ is inconsistent $KB_I \cup KB_{IC}$ with α.*
*(KB*7.3)* Weak relevance: *If $\beta \in KB \backslash KB * \alpha$, then there is a set KB' such that $KB' \subseteq KB \cup \alpha$, KB' is consistent $KB_I \cup KB_{IC}$ with α, but $KB' \cup \{\beta\}$ is inconsistent $KB_I \cup KB_{IC}$ with α.*

To revise KB by α, only those informations that are relevant to α in some sense can be added (as example in the introduction illustrates). ($KB * 7.1$) is very strong axiom allowing only minimum changes, and certain rational revision can not be carried out. So, relaxing this condition (example with more details can be found in (Aravindan 1995, Hansson, SO 1997a, Ferme & Hansson 2011 and Falappa, MA et al 2012), this can be weakened to relevance. ($KB * 7.2$) is relevance policy that still can not permit rational revisions, so we need to go next step. With ($KB * 7.3$) the relevance axiom is further weakened and it is referred to as ”core-retainment”.

5.1 Revision function

Suppose that we want to revise Horn knowledge base KB with respect to a single clause without using negation. We may construct revision using generalizing techniques from classical belief (base) revision (Falappa et al. 2012). Partial meet revision operator is syntax dependent and based on the foundational approach. In order to define it, first we need to define α-consistent-remainders.

Definition 11 (Remainder Set). *Let a Horn knowledge base KB be a set of Horn formulae, where α is Horn clause. The α-consistent-remainders of KB, noted by $KB \downarrow_\top \alpha$, is the set of KB' such that:*

1. *$KB' \subseteq KB$, ensuring that $KB_I \subseteq KB'$ and $KB_{IC} \subseteq KB'$.*
2. *$KB' \cup \alpha$ is consistent with $KB_I \cup KB_{IC}$.*
3. *For any KB” such that $KB' \subset KB'' \subseteq KB$ then $KB'' \cup \alpha$ is inconsistent with $KB_I \cup KB_{IC}$.*

That is, $KB \downarrow_\top \alpha$ is the set of maximal KB-subsets consistent with α.

Example 2. Suppose that KB=$\{KB_I : p \leftarrow a \wedge b, p \leftarrow a, q \leftarrow a \wedge b;\ KB_U : a \leftarrow,$ $b \leftarrow;\ KB_{IC} : \emptyset\}$ and $\alpha = \leftarrow p$. Then we have that:
- $KB \downarrow_\top \alpha = \{\{p \leftarrow a \wedge b\}, \{p \leftarrow a\}, \{\leftarrow a\}, \{\leftarrow b\}\}$.

Revision by a Horn clause is based on the concept of a α-consistent-remainders. In order to complete the construction, we must define a selection function that selects consistent remainders.

5.2 Principle of minimal change

Let a Horn knowledge base KB be a set of Horn formulae and ψ is a Horn clause such that $KB = \{\phi \mid \psi \vdash \phi\}$ is derived by ϕ. Now we consider the revision of a Horn clause α wrt KB, that is $KB * \alpha$.

The principle of minimal change (PMC) leads to the definition of orders between interpretations. Let $\mathcal{I}$ be the set of all the interpretations and $Mod(\psi)$ be the set of models of ψ. A pre-order on $\mathcal{I}$, denoted $\leq_\psi$ is linked with ψ. The relation $<_\psi$ is defined from $\leq_\psi$ as usual:

$$I <_\psi I' \text{ iff } I \leq_\psi I' \text{ and } I' \nleq_\psi I.$$

The pre-order $\leq_\psi$ is *faithful* to ψ if it verifies the following conditions:

1) If $I, I' \in Mod(\psi)$ then $I <_\psi I'$ does not hold;
2) If $I \in Mod(\psi)$ and $I' \notin Mod(\psi)$ then $I <_\psi I'$ holds;
3) if $\psi \equiv \phi$ then $\leq_\psi = \leq_\phi$.

A minimal interpretation may thus be defined by:

$\mathcal{M} \subseteq \mathcal{I}$, the set of minimal interpretations in $\mathcal{M}$ according to $\leq_\psi$ is denoted $Min(\mathcal{M}, \leq_\psi)$. And I is minimal in $\mathcal{M}$ according to $\leq_\psi$, if $I \in \mathcal{M}$ and there is no $I' \in \mathcal{M}$ such that $I' <_\psi I$.

Revision operation * satisfies the postulates (KB*1) to (KB*6) and (KB*7.3) if and only if there exists a total pre-order $\leq_\psi$ such that:

$$Mod(\psi * \phi) = Min(Mod(\phi), \leq_\psi).$$

Definition 12 (Selection function). *Let KB be a Horn knowledge base. γ is a selection function for KB iff for all Horn clauses α*

1. *If $KB \downarrow_\top \alpha \neq \emptyset$ then $\emptyset \neq \gamma(KB \downarrow_\top \alpha) \subseteq KB \downarrow_\top \alpha$.*
2. *If $KB \downarrow_\top \alpha = \emptyset$ then $\gamma(KB \downarrow_\top \alpha) = \{KB\}$*

Observation 1 *Let KB, KB' be an Horn knowledge base, KB' be consistent. Suppose that $\alpha \in KB$ and $\alpha \in KB'$. Then $\alpha \in X$ for all $X \in KB \downarrow_\top KB'$ and, therefore, $\alpha \in \bigcap(KB \downarrow_\top KB')$.*

From the above observation and definition 11 it follows that all the Horn knowledge base of $KB \cap \alpha$ are "protected", in the sense that they are included in the intersection of any collection of remainders. That is, a consolidated selection function selects a subset of the set $KB \downarrow_\top \alpha$ whose elements all contain the set $KB \cap \alpha$.

Definition 13 (Partial meet revision). *Let KB be a Horn knowledge base with an immutable part KB_I and γ a selection function for KB. The partial meet revision on KB that is generated by γ is the operator $*_\gamma$ such that, for all Horn clauses α:*

$$KB *_\gamma \alpha = \begin{cases} \cap\gamma(KB \downarrow_\top \alpha) \cup \alpha & \textit{if } \alpha \textit{ is consistent with } KB_I \cup KB_{IC} \\ KB & \textit{otherwise.} \end{cases}$$

An operator * is a generalized revision (partial meet revision) on KB if and only if there is a selection function γ for KB such that for all Horn clauses α, KB*α = KB$*_\gamma\alpha$.

Example 3. Given KB={$KB_I : p \leftarrow a \wedge b, p \leftarrow a, q \leftarrow a \wedge b$; $KB_U : a \leftarrow, b \leftarrow$; $KB_{IC} : \emptyset$}, α= $\leftarrow p$ and $KB \downarrow_\top \alpha$= {{$p \leftarrow a \wedge b$}, {$p \leftarrow a$}, {$\leftarrow a$}, {$\leftarrow b$}}. We have two possible results for the selection function and its associated partial meet revision operator

$$\gamma_1(KB \downarrow_\top \alpha) = \{\leftarrow p\} \text{ and } KB *_{\gamma_1} \alpha = \{p \leftarrow a \wedge b, \leftarrow a, \leftarrow b\}$$
$$\gamma_2(KB \downarrow_\top \alpha) = \{\leftarrow p\} \text{ and } KB *_{\gamma_2} \alpha = \{p \leftarrow a, \leftarrow a\}$$

The partial meet revision on KB that is generated by γ_2 gives minimal interpretation with respect to PMC.

Theorem 5. *For every Horn knowledge base KB, $*$ is a generalized revision function iff it satisfies the postulates (KB*1) to (KB*6) and (KB*7.3).*

Proof.

(**If part**) * satisfies (KB*1) to (KB*6) and (KB*7.3). We must show that $*$ is a generalized revision. When $KB_I \vdash \alpha$, (KB*1) to (KB*6) and (KB*7.3) imply that $KB * \alpha = KB$ is coinciding with generalized revision.

When $KB_I \vdash \neg\alpha$, the required result follows from the two observations:

1. $\exists KB' \in KB \downarrow_\top \alpha$ s.t.$KB * \alpha \subseteq KB'$ (when $KB * \alpha = KB \cup \{\alpha\}$)
   Let γ be an selection function for KB and $*_\gamma$ be the generalized revision on KB that is generated by γ. Since * satisfies closure (KB*1), $KB *_\gamma \alpha$ is KB contained in α. Also, satisfaction of weak success postulate (KB*2) ensures that $\alpha \subseteq KB{*_\gamma}\alpha$. Every element of $KB \downarrow_\top \alpha$ is a inclusion maximal subset that does drive α, and so any subset of KB that does derive α must be contained in a member of $KB \downarrow_\top \alpha$.
2. $\bigcap(KB \downarrow_\top \alpha) \subseteq KB *_\gamma \alpha$ (when $KB * \alpha = KB \cup \{\alpha\}$)
   Consider any $\beta \in \bigcap(KB \downarrow_\top \alpha)$. Assume that $\beta \notin KB * \alpha$. Since * satisfies weak relevance postulate (KB*7.3), it follows that there exists a set KB' s.t. $KB' \subseteq KB \cup \alpha$; KB' is consistent with α; and $KB' \cup \{\beta\}$ is inconsistent with α. But this contradicts the that β is present in every maximal subset of KB that does derive α. Hence β must not be in $KB *_\gamma \alpha$.

(**Only if part**) Let $KB * \alpha$ be a generalized revision of α for KB. We have to show that $KB * \alpha$ satisfies the postulate (KB*1) to (KB*6) and (KB*7.3).

Let γ be an selection function for KB and $*_\gamma$ be the generalized revision on KB that is generated by γ.

Closure Since $KB *_\gamma \alpha$ is a Horn knowledge base, this postulate is trivially shown.

Weak Success Suppose that α is consistent. Then it is trivial by definition that $\alpha \subseteq KB *_\gamma \alpha$.

Inclusion Since every $X \in KB \downarrow_\top \alpha$ is such that $X \subseteq KB$ then this postulate is trivially shown.

Immutable-inclusion Since every $X \in KB \downarrow_\top \alpha$ is such that $X \subseteq KB_I$ then this postulate is trivially shown.

Vacuity 1 Trivial by definition.

Vacuity 2 If $KB \cup \{\alpha\}$ is consistent then $KB \downarrow_\top \alpha = \{\{\text{KB}\}\}$. Hence $KB *_\gamma \alpha = KB \cup \{\alpha\}$.

Consistency Suppose that α is consistent. Then $KB \downarrow_\top \alpha \neq= \emptyset$ and by definition, every $X \in KB \downarrow_\top \alpha$ is consistent with α. Therefore, the intersection of any subset of $KB \downarrow_\top \alpha$ is consistent with α. Finally, $KB *_\gamma \alpha$ is consistent.

Uniformity If α and β are KB-equivalent, then $KB \downarrow_\top \alpha = KB \downarrow_\top \beta$

Weak relevance Suppose that $KB \downarrow_\top \alpha \neq \emptyset$. Let $\beta \in KB$. Then there is some $X \in KB \downarrow_\top \alpha$ such that $\beta \notin X$. Therefore, there is some X such that $\beta \notin X \subseteq KB$, $X \cup \alpha$ is consistent but $X \cup \alpha \cup \{\beta\}$ is inconsistent.
Suppose that $KB \downarrow_\top \alpha = \{\emptyset\}$ in which case α is inconsistent. By definition, $KB *_\gamma \alpha = KB$ and weak relevance is vacuously satisfied.

5.3 Horn knowledge base revision with hitting set

In this section, we show that Horn knowledge base revision has an interesting connection with kernel change via hitting set.

Kernel revision system

To revise a Horn formula α from a Horn knowledge base KB, the idea of kernel revision is to *keep at least* one element from every inclusion-minimal subset of KB that derives α. Because of the immutable-inclusion postulate, no Horn formula from KB_I can be deleted.

Definition 14 (Kernel sets). *Let a Horn knowledge base KB be a set of Horn formulae, where α is Horn clause. The α-inconsistent kernel of KB, noted by $KB \bot_\bot \alpha$, is the set of KB' such that:*

1. *$KB' \subseteq KB$ ensuring that $KB_I \subseteq KB'$ and $KB_{IC} \subseteq KB'$.*
2. *$KB' \cup \alpha$ is inconsistent with $KB_I \cup KB_{IC}$.*
3. *For any KB" such that $KB'' \subset KB' \subseteq KB$ then $KB'' \cup \alpha$ is consistent with $KB_I \cup KB_{IC}$.*

That is, given a consistent α, $KB\perp_{\perp}\alpha$ is the set of minimal KB-subsets inconsistent with α.

Example 4. Suppose that KB=$\{KB_I : p \leftarrow a \wedge b, p \leftarrow a, q \leftarrow a \wedge b;\ KB_U : a \leftarrow, b \leftarrow;\ KB_{IC} : \emptyset\}$ and $\alpha = \leftarrow p$. Then we have that:
$KB\perp_{\perp}\alpha = \{\{p \leftarrow a \wedge b\}, \{p \leftarrow a\}\}$.

Revision by a Horn clause is based on the concept of a α-inconsistent-kernels. In order to complete the construction, we must define a incision function that cuts in each inconsistent-kernel.

Definition 15 (Incision function). *Let KB be a set of Horn formulae. σ is a incision function for KB if and only if, for all consistent Horn clauses α*

1. $\sigma(KB\perp_{\perp}\alpha) \subseteq \bigcup KB\perp_{\perp}\alpha$
2. *If $KB' \in KB\perp_{\perp}\alpha$ then $KB' \cap (\sigma(KB\perp_{\perp}\alpha)) \neq 0$*

Definition 16 (Hitting set). *A* hitting set *H for $KB\perp_{\perp}\alpha$ is defined as a set s.t. (i) $H \subseteq \bigcup(KB\perp_{\perp}\alpha)$, (ii) $H \cap KB_I$ is empty and (iii) $\forall X \in KB\perp_{\perp}\alpha$, $X \neq \emptyset$ and $X \cap KB_U$ is not empty, then $X \cap H \neq \emptyset$.*

A hitting set is said to be *maximal* when H consists of all updatable statements from $\bigcup(KB\perp_{\perp}\alpha)$ and *minimal* if no proper subset of H is a hitting set for $KB\perp_{\perp}\alpha$.

Observation 2 *Let KB, KB' be an Horn knowledge base, KB' be consistent. Suppose that $\alpha \in KB$ and $\alpha \in KB'$. Then $\alpha \notin \bigcup(KB\perp_{\perp}KB')$ and, therefore, $KB' \cap \bigcup(KB\perp_{\perp}KB') = 0$*

From the above observation and definition 15 it follows that all the Horn knowledge base of α are "protected", in the sense that they can not be considered for removing by the consolidated incision function. That is, a consolidated incision function selects among the sentences of $KB\backslash\alpha$ that make $KB \cup \alpha$ inconsistent.

Definition 17 (Generalized Kernel revision). *An incision function for KB is a function s.t. for all α, $\sigma(KB\perp_{\perp}\alpha)$ is a hitting set for $KB\perp_{\perp}\alpha$. Generalized kernel revision on KB that is generated by σ is the operator $*_\sigma$ such that, for all Horn clauses α:*

$$KB *_\sigma \alpha = \begin{cases} (KB\backslash\sigma(KB\perp_{\perp}\alpha) \cup \alpha & \text{if } \alpha \text{ is consistent } KB_I \cup KB_{IC} \\ KB & \text{otherwise.} \end{cases}$$

An operator $*$ is a generalized kernel revision for KB if and only if there is an incision function σ for KB such that for all Horn clauses α, $KB * \alpha = KB *_\sigma \alpha$.

From the definition of hitting set, it is clear that when $KB \vdash \neg\alpha$, α is the hitting set of $KB\perp_{\perp}\alpha$. On the other hand, when $KB_I \vdash \alpha$, the definition ensures that only updatable elements are inserted, and α does follow from the revision. Thus, weak success (KB*2), immutable-inclusion (KB*3.2) and vacuity (KB*4.1) are satisfied by generalized kernel revision of α from KB.

Example 5. Given KB=$\{KB_I : p \leftarrow a \wedge b, p \leftarrow a, q \leftarrow a \wedge b;\ KB_U : a \leftarrow, b \leftarrow;\ KB_{IC} : \emptyset\ \}$, $\alpha = \leftarrow p$ and $KB \perp_\perp \alpha = \{\{p \leftarrow a \wedge b\}, \{p \leftarrow a\}\}$. We have two possible results for the incision function and its associated kernel revision operator:

$$\sigma_1(KB \perp_\perp \alpha) = \{p \leftarrow a \wedge b\} \text{ and } KB *_{\sigma_1} \alpha = \{\{\leftarrow a\}, \{\leftarrow b\}\},$$
$$\sigma_2(KB \perp_\perp \alpha) = \{p \leftarrow a\} \text{ and } KB *_{\sigma_2} \alpha = \{\{\leftarrow a\}\}.$$

Incision function σ_2 produces minimal hitting set for $KB \perp_\perp \alpha$.

Theorem 6. *For every Horn knowledge base KB, $*_\sigma$ is a generalized kernel revision function iff it satisfies the postulates (KB*1) to (KB*6) and (KB*7.3).*

Proof.

(**If part**) * satisfies (KB*1) to (KB*6) and (KB*7.3). We must show that $*$ is a generalized kernel revision. Let σ be a incision function and α Horn formula. When $KB_I \vdash \alpha$, (KB*1) to (KB*6) and (KB*7.3) imply that $KB * \alpha = KB$ coincides with generalized revision and follow PMC.

When $KB_I \vdash \neg\alpha$, the required result follows from the two observations:

1. $\exists KB' \in KB \perp_\perp \alpha$ s.t.$KB * \alpha \subseteq KB'$ (when $KB_I \vdash \alpha$)
Let σ be an incision function for KB and $*_\sigma$ be the generalized revision on KB that is generated by σ. Since * satisfies closure (KB*1), $KB *_\sigma \alpha$ is KB contained in α. Also, satisfaction of weak success postulate (KB*2) ensures that $\alpha \subseteq KB *_\sigma \alpha$. Every element of $KB \perp_\perp \alpha$ is a inclusion minimal subset that does derive α, and so any subset of KB that does derive α must be contained in a member of $KB \perp_\perp \alpha$.
2. $\bigcap(KB \perp_\perp \alpha) \subseteq KB *_\sigma \alpha$ (when $KB_I \vdash \alpha$)
Consider any $\beta \in \bigcap(KB \perp_\perp \alpha)$. Assume that $\beta \notin KB * \alpha$. Since * satisfies weak relevance postulate (KB*7.3), it follows that there exists a set KB' s.t. $KB' \subseteq KB \cup \alpha$; KB' is a consistent with α; and $KB' \cup \{\beta\}$ is inconsistent with α. But this contradicts that β is present in every minimal subset of KB that does derive α. Hence β must not be in $KB *_\sigma \alpha$.

(**Only if part**) Let $KB * \alpha$ be a generalized revision of α for KB. We have to show that $KB * \alpha$ satisfies the postulate (KB*1) to (KB*6) and (KB*7.3).

Let σ be an incision function for KB and $*_\sigma$ be the generalized revision on KB that is generated by σ.

Closure Since $KB *_\sigma \alpha$ is a Horn knowledge base, this postulate is trivially shown.

Weak Success Suppose that α is consistent. Then it is trivial by definition that $\alpha \subseteq KB *_\sigma \alpha$.

Inclusion Trivial by definition.

Immutable-inclusion Since every $X \in KB \perp_\perp \alpha$ is such that $X \subseteq KB_I$ then this postulate is trivially shown.

Vacuity 1 Trivial by definition.

Vacuity 2 If $KB \cup \{\alpha\}$ is consistent then $KB \perp_{\perp} \alpha = \{\{\text{KB}\}\}$. Hence $KB *_{\sigma} \alpha$ $= KB \cup \{\alpha\}$.

Consistency Suppose that α is consistent. Then $KB \perp_{\perp} \alpha \neq= \emptyset$ and by definition, every $X \in KB \perp_{\perp} \alpha$ is consistent with α. Therefore, the intersection of any subset of $KB \perp_{\perp} \alpha$ is consistent with α. Finally, $KB *_{\sigma} \alpha$ is consistent.

Uniformity If α and β are KB-equivalent, then $KB \perp_{\perp} \alpha = KB \perp_{\perp} \beta$

Weak relevance Let $\beta \in KB$ and $\beta \notin KB *_{\sigma} \alpha$. Then $KB *_{\sigma} \alpha \neq KB$ and, from the definition of $*_{\sigma}$,it follows that:

$$KB *_{\sigma} \alpha{=}(KB \backslash \sigma(KB \perp_{\perp} \alpha)) \cup \alpha$$

Therefore, from $\beta \notin (KB \backslash \sigma(KB \perp_{\perp} \alpha)) \cup \alpha$ and $\beta \in KB$, we can conclude that $\beta \in \sigma(KB \perp_{\perp} \alpha)$. By definition $\sigma(KB \perp_{\perp} \alpha) \subseteq \bigcup KB \perp_{\perp} \alpha$, and it follows that there is some $X \in KB \perp_{\perp} \alpha$ such that $\beta \in X$. X is a minimal KB-subset inconsistent with α. Let $Y = X \backslash \{\beta\}$. Then Y is such that $Y \subset X \subseteq KB \subseteq KB \cup \alpha$. Y is consistent with α but $Y \cup \{\beta\}$ is consistent with α.

From the Theorem 5 and 6, it immediately follows that a revision operation on a Horn knowledge base is a generalized kernel revision iff it is a generalized revision. The following theorem formalizes this with additional insights into the relationship between kernel and generalized revisions.

Theorem 7.

1. *A revision operation over a Horn knowledge base KB is a generalized kernel revision over KB iff it is a generalized revision over KB.*
2. *When the incision function σ is minimal, i.e. the hitting set defined by σ is inclusion-minimal, then the generalized kernel revision defined by σ is a partial meet revision of α from KB.*
3. *When the incision function σ is maximal, i.e. $\sigma(KB \perp_{\perp} \alpha)$ consists of all updatable statements from $\bigcup(KB \perp_{\perp} \alpha)$, then the kernel contraction defined by σ is the minimal generalized revision of α from KB.*

Proof. Follows from the Definition 16, Theorem 5 and 6

6 Knowledge base dynamics and abduction

In this Section, we study the relationship between Horn knowledge base dynamics, discussed in the previous Section, and abduction that was introduced by the philosopher Peirce (see Aliseda 2006, Boutilier & Beche 1995 and Pagnucco 1996). We show how an abduction grammar could be used to realize revision with an immutability condition. A special subset of literals (atoms) of language $\mathcal{L}_{\mathcal{H}}$, *abducibles* Ab, are designated for abductive reasoning. Our work is based on atoms (literals), so we combine Christiansen and Dahl (Christiansen & Dahl 2009) grammars approach. Simply, we want to compute abducibles for Horn knowledge base (Horn or Horn logic with stratified negation).

Example 6. Consider a Horn logic with stratified negation knowledge base KB with immutable part KB_I, updatable part KB_U and integrity constraint KB_{IC}.

$$
\begin{array}{llll}
KB_I : & flies(x) \leftarrow bird(x), not\ ab(x), & KB_U : & bird(tweety) \leftarrow \qquad KB_{IC} : \emptyset \\
 & ab(x) \leftarrow broken\_wing(x) & & bird(opus) \leftarrow \\
 & & & broken\_wing(tweety) \leftarrow
\end{array}
$$

If we observe that tweety flies, there is a good reason to assume that the wound has already healed. Then, removing the fact broken_wing(tweety) from the KB explains the observation flies(tweety). On the other hand, suppose that we later notice that opus does not fly anymore. Since flies(opus) is entailed by KB_I, we now have to revise the Horn knowledge base to block the derivation of flies(opus) by assuming, for instance, broken_wing(opus). In nonmonotonic theories, deletion of formulae may introduce new formulae, thus positive (Δ^+) and negative (Δ^-) explanations play a complementary role in accounting for an observation in nonmonotonic theories. (more explanation in Sakama & Inoue 2003)

Definition 18 (Abductive grammar). *A abductive grammar Γ is a 6-tuple $\langle N, T, IC,$ $KB, R, S\rangle$ where:*

- *N are nonterminal symbols in the immutable part (KB_I).*
- *T is a set of terminal symbols in the updatable part (KB_U).*
- *IC is the set of integrity constraints for the Horn knowledge base (KB_{IC}).*
- *KB is the Horn knowledge base which consists of $KB = KB_I \cup KB_U \cup KB_{IC}$.*
- *R is a set of rules, $R \subseteq KB$.*
- *S is the revision of literals (atoms), called the start symbol.*

Example 7. Consider a Horn knowledge base KB (with immutable part KB_I, updatable part KB_U and integrity constraint KB_{IC}) and a Horn clause α (p is α) be revise.

$$
\begin{array}{llllll}
KB_I : & p \leftarrow q \wedge a & KB_U : & a \leftarrow & KB_{IC} : & \leftarrow b \\
 & p \leftarrow r \wedge b & & r \leftarrow & & \\
 & q \leftarrow c \wedge d & & & & \\
 & r \leftarrow e \wedge f & & & & \\
 & p \leftarrow b & & & &
\end{array}
$$

KB be a Horn knowledge base, represented by the grammar ($\Gamma = \langle N, T, KB, R, S\rangle$) as follows:

N={p}
T={a,b,c,d,e,f,q,r}
IC={b}
KB=$KB_I \cup KB_U \cup KB_{IC}$
R={$p \leftarrow q, a;\ p \leftarrow r, b;\ q \leftarrow c, d;\ r \leftarrow e, f;\ p \leftarrow b;\ a;\ r$}

S={p}

Definition 19 (Constraint system). *A constraint system for abduction is a pair* $\langle KB^{Ab},$
$KB^{BG}\rangle$*, where* $KB^{Ab}(\Delta)$ *is a set of propositions (abducibles) and* KB^{BG} *a background Horn knowledge base.*

Note 2. In the sequel, without any loss of generality, we assume that KB_I is a set of rules and KB_U is a set of abducibles from Horn knowledge base perspective. With respect to the considered grammars, KB^{BG} is a set all Horn formulae from R and KB^{Ab} is set of abducibles from T.

Note 3. Given a Horn knowledge base KB and a Horn clause α, the problem of abduction is to explain α in terms of an abduction, i.e. to generate a set of abducibles KB^{Ab}, Δ s.t. $KB^{BG} \cup \Delta \vdash \alpha$.

Definition 20 (Minimal abductive explanation). *Let KB be a Horn knowledge base and* α *an observation to be explained. Then, for a set of abducibles* (KB^{Ab})*,* Δ *is said to be an abductive explanation with respect to* KB^{BG} *iff* $KB^{BG} \cup \Delta \vdash \alpha$*.* Δ *is said to be* minimal *with respect to* KB^{BG} *iff no proper subset of* Δ *is an abductive explanation for* α*, i.e.* $\nexists \Delta^{'}$ *s.t.* $KB^{BG} \cup \Delta^{'} \vdash \alpha$*.*

Since an incision function is adding and removing only updatable elements from each member of the kernel set, to compute a generalized revision of α from KB, we need to compute only the abduction in every α-kernel of KB. So, it is now necessary to characterize precisely the abducibles present in every α-kernel of KB. The notion of minimal abductive explanation is not enough to capture this, and we introduce locally minimal and KB-closed abductive explanations explanations.

Definition 21 (Local minimal abductive explanations). *Let* $KB^{BG'}$ *be a subset of* KB^{BG}*, s.t* Δ *is a minimal abductive explanation of* α *with respect to* $KB^{BG'}$ *(for some* Δ*). Then* Δ *is called local minimal for* α *with respect to* KB^{BG}*.*

Example 8. From example 7, suppose $\{p \leftarrow q \wedge a, p \leftarrow a\}$, where a and f are abducibles in the grammar system R. Clearly, $\Delta_1 = \{a\}$ is the only minimal abductive explanation for p with respect to R. $\Delta_2 = \{a, q\}$ is an abductive explanation for p with respect to R, but not a minimal one. However, Δ_2 is a locally minimal abductive explanation for p with respect to R, since it is a minimal explanation for p with respect to $\{p \leftarrow q \wedge a\}$ which is a subset of R.

The concept of locally minimal abductive explanation is computationally attractive, since minimal abductive explanation is more expensive to compute (Aravindan 1995). To find a minimal admissible and denial literal (atom) from KB^{Ab} that is positive and negative literal (atom) from KB^{Ab}, we need to introduce a constraint system (C) with integrity constraint (IC).

Definition 22 (Constraint abduction system). *A constrained abductive grammar is a pair $\langle \Gamma, C \rangle$, where Γ is an abductive grammar and C a constraint system for abduction, $\Gamma = \langle N, T, R, S \rangle$ and $C = \langle KB^{BG}, KB^{Ab}, IC \rangle$.*

Given a constrained abductive grammar $\langle \Gamma, C \rangle$ as above, the constrained abductive recognition problem for $\tau \in T^*$ is the problem of finding an admissible and denial knowledge base (Horn knowledge base contained set of positive and negative literal (atoms)) from KB^{Ab} and such that $\tau \in \mathcal{L}_P(\Gamma_{KB^{Ab}})$ where $\mathcal{L}_P(\Gamma_{KB^{Ab}})$ is propositional language over abducibles in Γ, where $\Gamma_{KB^{Ab}} = \langle N, T, KB^{BG} \cup KB^{Ab}, R, S \rangle$. In this case, KB^{Ab} is called a *constrained (abductive) system of* τ. Such that KB^{Ab} is minimal whenever no proper subset of it is in τ given $\langle \Gamma, C \rangle$.

Example 9. We extend example 7, in order to show that C is constraint system C, with C $= \langle KB^{BG}, KB^{Ab}, IC \rangle$

$KB^{BG} = \{p \leftarrow q, a;\ p \leftarrow r, b;\ q \leftarrow c, d;\ r \leftarrow e, f;\ p \leftarrow b;\ a;\ r\}$
$KB^{Ab} = \{a, b, c, d, e, f, q, r\}$
IC $= \{\leftarrow b\}$

Note 4. Let $KB^{Ab} \in (\{\Delta^+, \Delta^-\})$. Here Δ^+ refers to admission Horn knowledge base (positive atoms) and Δ^- refers to denial Horn knowledge base (negative atoms) with respect to given α. The abduction problem is to explain Δ with abducibles (KB^{Ab}), s.t. $KB^{BG} \cup \Delta^+ \cup \Delta^- \models \alpha$ and $KB^{BG} \cup \Delta^+ \models \alpha \cup \Delta^-$ are both consistent with IC.

An admission and denial Horn knowledge base, based on $\langle KB^{BG}, KB^{Ab} \rangle$ is a set KB^{Ab} of atoms (literals) whose propositions are in KB^{Ab} such that $KB^{BG} \cup KB^{Ab}$ is consistent with IC.

Example 10. From Example 9 and Note 4, the constraint system C, with C $= \langle KB^{BG}, KB^{Ab}, IC \rangle$

$KB^{BG} = \{p \leftarrow q, a;\ p \leftarrow r, b;\ q \leftarrow c, d;\ r \leftarrow e, f;\ p \leftarrow b;\ a;\ r\}$
$KB^{Ab} = \{\Delta^+ = \{a, c, d, e, f, q, r\}$ and $\Delta^- = \{a, b, r\}\}$
IC $= \{\leftarrow b\}$

Definition 23 (KB-closed abductive explanations). *For a set of abducibles (KB^{Ab}), Δ^+ and Δ^- are said to be closed abductive explanations with respect to KB^{BG} iff $KB^{BG} \cup \Delta^+ \cup \Delta^- \models \alpha$ and $KB^{BG} \cup \Delta^+ \models \alpha \cup \Delta^-$. Δ^+ and Δ^- are said to be* minimal *with respect to KB^{BG} iff no proper subset of Δ^+ and Δ^- is an abductive explanation for α, i.e. $\nexists \Delta^{+'} \subsetneq \Delta^+$ and $\nexists \Delta^{-'} \subsetneq \Delta^-$ s.t. $KB^{BG} \cup \Delta^{+'} \cup \Delta^{-'} \models \alpha$ and $KB^{BG} \cup \Delta^{+'} \models \alpha \cup \Delta^{-'}$ both consistent with IC.*

KB-closed abductive explanations are also known as KB-closed local minimal explanations.

Observation 3 *Let $KB^{BG'}$ be a smallest subset of KB^{BG} s.t, Δ^+ and Δ^- minimal abductive explanations of α with respect to $KB^{Ab'}$ and $KB^{BG'}$ (for some Δ^+ and Δ^-). Then Δ^+ and Δ^- are called* locally minimal *for α with respect to $KB^{Ab'}$ and KB^{BG} and consistent with IC.*

Example 11. $\Delta^+ = \{a, c, d\}$ and $\Delta^- = \{c\}$ with respect to IC are only locally minimal abductive explanations for p with respect to $KB^{BG'}$ (more explanations can be found in (Lu W 1999)).

From example 10 and definition 23, we want to show that the constraint system C, with C $=\langle KB^{BG}, KB^{Ab}, IC\rangle$

$KB^{BG} = \{p \leftarrow q, a;\ q \leftarrow c, d;\ a;\ c;\ d\}$
$KB^{Ab} = \{\Delta^+ = \{a, c, d\}$ and $\Delta^- = \{b, r\}\}$ and IC $=\{\leftarrow b\}$

Now, we need to connect the grammar system Γ to the Horn (stratified) knowledge base KB, such that $KB_I \cup KB_U \cup KB_{IC} = KB^{BG} \cup KB^{Ab} \cup IC$ holds. The connection between locally minimal abductive explanation for α with respect to KB_I and α-kernel of KB, which is shown by the following lemma immediately follows from their respective definitions.

Observation 4

1. *Let KB be a Horn (stratified) knowledge base and α a Horn clause s.t. $\nvDash \neg\alpha$. Let Δ^+ and Δ^- be a KB-closed locally minimal abductive explanation for α with respect to KB_I. Then, there exists an α-kernel X of KB s.t. $X \cap KB_U = \Delta^+ \cup \Delta^-$.*
2. *Let KB be a Horn (Horn logic with stratified negation) knowledge base and α a Horn clause s.t. $\nvDash \neg\alpha$. Let X be a α-kernel of KB and $\Delta^+ \cup \Delta^- = X \cap KB_U$. Then, Δ^+ and Δ^- are KB - closed locally minimal abductive explanations for α with respect to KB_I.*

Proof.

1. The fact that $\nvDash \neg\alpha$ and there exists a KB - closed locally minimal abductive explanation for α with respect to KB_I, it is clear that there exists at least one α- kernel of KB. Suppose Δ ($\Delta \in \Delta^+ \cup \Delta^-$) is empty (i.e. $KB_I \models \neg\alpha$), then the required result follows immediately. If not, since Δ is a locally minimal abductive explanation, there exists a minimal subset $KB'_I \subseteq KB_I$, s.t. Δ is minimal abductive explanation of α with respect to KB'_I. Since, Δ is KB-closed, it is not difficult to see that $KB'_I \cup \Delta^+ \cup \Delta^-$ is a α - kernel of KB.
2. Since X is a α - kernel of KB and Δ is the set of all abducibles in X, it follows that $\Delta^+ \cup \Delta^-$ is a minimal abductive explanation of Δ with respect to $X \backslash \Delta^- \cup \Delta^+$. It is obvious that $\Delta^+ \cup \Delta^-$ is KB- closed, and so Δ is a KB-closed locally minimal abductive explanation for α with respect to KB_I.

Theorem 8. *Consider a constrained abductive grammar $AG = \langle \Gamma, C \rangle$ with $\Gamma = \langle N, T, KB, R, S \rangle$ and $C = \langle KB^{BG}, KB^{Ab}, IC \rangle$. Construct a abductive grammar $\Delta(AG) = \langle N, T, KB^{BG}, R, S \rangle$ by having, for any (Δ^{+}) (or) (Δ^{-}) from KB^{Ab}, the set of acceptable results for accommodate $(\alpha, KB^{BG} \in \Delta^{+})$ being of the form $(KB^{Ab} \backslash \Delta^{+})$ where $(\Delta^{+} \in KB^{Ab'})$. Δ^{+} is a locally minimal set of atoms (literals) $KB^{BG} \cup \Delta^{+}$ and $KB^{BG} \cup \Delta^{+} \models \alpha$ is consistent with IC; if (Δ^{-}) exists procedure is similar, (like denial (Δ^{-}) being of the form $(KB^{Ab} \backslash \Delta^{-})$. Δ^{-} is a locally minimal set of atoms (literals) $KB^{BG} \cup \Delta^{-}$ and $KB^{BG} \cup \Delta^{-} \models \alpha$ is consistent with IC), otherwise accommodate $(\alpha, KB^{BG} \in \Delta^{-})$ is not possible.*

Proof. From Observation 3, Let $KB^{BG'}$ be a smallest subset of KB^{BG} s.t, Δ^{+} and Δ^{-} minimal abductive explanations of α with respect to $KB^{Ab'}$ and $KB^{BG'}$ (for some Δ^{+} and Δ^{-}). Then Δ^{+} and Δ^{-} are called *locally minimal* for α with respect to $KB^{Ab'}$ and KB^{BG} and consistent with IC.

From Observation 4, (Δ^{-}) is follow to the kernel of KB and Δ is the set of all abducibles in $(\alpha, KB^{BG} \in \Delta^{+})$, it follows that $\Delta^{+} \cup \Delta^{-}$ is a minimal abductive explanation of Δ with respect to $KB \backslash \Delta^{-} \cup \Delta^{+}$. It is obvious that $\Delta^{+} \cup \Delta^{-}$ is KB- closed, and so Δ is a KB-closed locally minimal abductive explanation for α with respect to KB_{I}.

(Δ^{-}) is not follow from KB - closed locally minimal abductive explanation for α with respect to KB_{I}, it is clear that there exists at least one α- kernel of KB.

An immediate consequence of the above observation 4 is that it is enough to compute all the KB-closed locally minimal abductive explanations for α with respect to KB_{I} in order to revise α from KB. Thus, a well-known abductive procedure to compute an abductive explanation for α with respect to KB_{I} could be used:

Theorem 9. *Let KB be a Horn (stratified) knowledge base and α a Horn clause.*

1. *If Algorithm 1 produces KB' as a result of revision α to KB, then KB' is a generalized revision of α from KB.*
2. *If KB' is a generalized revision of α from KB, then there exists an incision function σ s.t. KB' is produced by Algorithm 1 as a result of revision α from KB, using σ.*

Proof. Follows from Observation 4 and Theorem 7

6.1 Generalized revision algorithm

The problem of Horn knowledge base revision is concerned with determining how a request to change can be appropriately translated into one or more atoms or literals. In this section we develop a new generalized revision algorithm. Note that it is enough to compute all the KB-locally minimal abduction explanations for α with respect to $KB_{I} \cup KB_{U} \cup KB_{IC}$. If α is consistent with KB then a well-known abductive procedure for compute an abductive explanation for α

with respect to KB_I could be used to compute kernel revision.

Algorithm 1 Generalized revision algorithm

```
Input :   A Horn knowledge base $KB = KB_I \cup KB_U \cup KB_{IC}$
          and a Horn clause $\alpha$ to be revised.
Output:   A new Horn knowledge base $KB' = KB_I \cup KB_U^* \cup KB_{IC}$,
          s.t. $KB'$is a generalized revision $\alpha$ to KB.
Procedure $KB(KB, \alpha)$
begin
  1.  Let V:= $\{c \in KB_{IC} \mid KB_I \cup KB_{IC}$ inconsistent with $\alpha$ with respect to $c\}$
      $P := N := \emptyset$ and $KB' = KB$
  2.  While $(V \neq \emptyset)$
        select a subset $V' \subseteq V$
        For each $v \in V'$, select a literal to be
        remove (add to N) or a literal to be added (add to P) with respect to KB
        Let KB := $KR(KB, P, N)$
        Let V:= $\{c \in KB_{IC} \mid KB_I$ inconsistent with $\alpha$ with respect to $c\}$
      return
  3.  Produce a new Horn knowledge base $KB'$
end.
```

```
Procedure $KR(KB, \Delta^+, \Delta^-)$
begin
  1.  Let $P := \{e \in \Delta^+ \mid KB_I \not\models e\}$ and $N := \{e \in \Delta^- \mid KB_I \models e\}$
  2.  While $(P \neq 0)$ or $(N \neq 0)$
        select a subset $P' \subseteq P$ or $N' \subseteq N$
        Construct a set $S_1 = \{X \mid X$ is a KB-closed locally
          minimal abductive wrt P explanation for $\alpha$ wrt $KB_I\}$.
        Construct a set $S_2 = \{X \mid X$ is a KB-closed locally
          minimal abductive wrt N explanation for $\alpha$ wrt $KB_I\}$.
  3.  Determine a hitting set $\sigma(S_1)$ and $\sigma(S_2)$
        If $((N = 0)$ and $(P \neq 0))$
            Produce $KB' = KB_I \cup \{(KB_U \cup \sigma(S_1)\}$
        else
            Produce $KB' = KB_I \cup \{(KB_U \backslash \sigma(S_2) \cup \sigma(S_1)\}$
        end if
        If $((N \neq 0)$ and $(P = 0))$
            Produce $KB' = KB_I \cup \{(KB_U \backslash \sigma(S_2)\}$
        else
            Produce $KB' = KB_I \cup \{(KB_U \backslash \sigma(S_2) \cup \sigma(S_1)\}$
        end if
  4.  return $KB'$
end.
```

Reasoning about Abduction

Definition 24 ((Teniente & Olive 1995)). *Let KB=(KB_I, KB_U, KB_{IC}) be a Horn knowledge base, T is updatable part from KB. We define the abduction framework $\langle KB^{BG}, KB^{Ab}, IC\rangle$. After Algorithm 1 is executed, u is derived part from KB'. The abduction explanation for u in $\langle KB_I \cup KB^*_U, KB_{IC}\rangle$ is any set T_i, where $T_i \subseteq KB^{Ab}$ such that: $KB_I \cup KB^*_U \cup T \models u$.*

An explanation T_i is minimal if no proper subset of T_i is also an explanation, i.e. if it does not exist any explanation T_j for u such that $T_j \subset T_i$

Reasoning about Deduction

Definition 25 ((Teniente & Olive 1995)). *Let KB=(KB_I, KB_U, KB_{IC}) be a Horn knowledge base, T is updatable part from KB. After Algorithm 1 is executed, u is derived part from KB'. The deduction consequence on u due to the application of T, $KB_I \cup KB^*_U \cup T \cup u$ is the answer to any question.*

Example 12. Consider a Horn knowledge base KB with immutable part KB_I, updatable part KB_U and integrity constraint KB_{IC}, compute closed local minimum with respect to to p.

$$\begin{array}{rlrlrl}
KB_I: & p \leftarrow q \wedge a & KB_U: & a \leftarrow & KB_{IC}: & \leftarrow b \\
& p \leftarrow r \wedge b & & r \leftarrow & & \\
& q \leftarrow c \wedge d & & & & \\
& r \leftarrow e \wedge f & & & & \\
& p \leftarrow b & & & &
\end{array}$$

From algorithm 1, the above example execute following steps:

Step number with execution

(Input) $KB_I: p \leftarrow q \wedge a, p \leftarrow r \wedge b, q \leftarrow c \wedge d, r \leftarrow e \wedge f, p \leftarrow b$
$KB_U: a \leftarrow, r \leftarrow$
$KB_{IC}: \leftarrow b$
(0) $\{p \leftarrow q, a;\ p \leftarrow r, b;\ q \leftarrow c, d;\ r \leftarrow e, f;\ p \leftarrow b;\ a;\ r\}$
(1) $\{V = b\}$
(2) $\{P = \{a, c, d, e, f, q, r\}$ and $N = \{a, r\}\}$
(2.1) $\{\Delta^+ = \{a, c, d, e, f, q, r\}$ and $\Delta^- = \{a, r\}\}$
(2.2) $\{\Delta^+ = \{a, c, d\}$ and $\Delta^- = \{\}\}$
(3) $\{p \leftarrow q, a;\ q \leftarrow c, d;\ a;\ c;\ d;\ r\}$
(Output) $KB_I: p \leftarrow q \wedge a, p \leftarrow r \wedge b, q \leftarrow c \wedge d, r \leftarrow e \wedge f, p \leftarrow a, p \leftarrow b$
$KB^*_U: a \leftarrow, c \leftarrow, d \leftarrow, r \leftarrow$
$KB_{IC}: \leftarrow b$

Theorem 10. *Let KB be a Horn knowledge base and α is (Horn or Horn logic with stratified negation) formula.*

1. *If Algorithm 1 produced KB' as a result of revising α from KB, then KB' satisfies all the rationality postulates (KB*1) to (KB*6) and (KB*7.3).*

2. *Suppose KB'' satisfies all these rationality postulates for revising α from KB, then KB'' can be produced by Algorithm 1.*

Proof. Follows from Theorem 6 and Theorem 9

7 Belief update VS Knowledge base update

In this section we give overview of how belief update is related to knowledge base update. This section is motivated by the works of Konieczny 2011 and Baral & Zhang 2005.

7.1 Belief revision vs Belief update

Intuitively, revision operators bring a minimal change to the base by selecting the most plausible models among the models of the new information. Whereas update operators Konieczny 2011 bring a minimal change to each possible world (model) of the base in order to take into account the change described by the new information, whatever the possible world.

Theorem 11 ([29]). *If $\circ$ is a revision operator (i.e. it satisfies (R1)-(R6)), then the update operator $\diamond$ defined by $\psi \diamond \mu = \bigvee_{w \models \psi} \psi_{\{w\}} \circ \mu$ is an update operator that satisfies (U1)-(U9).*

This theorem states that update can be viewed as a kind of pointwise revision.

7.2 Knowledge base revision vs Knowledge base update

Generalized revision algorithm brings principle of minimal change, according to new information; how a request to change Horn knowledge base can be appropriately translated into one or more literals. Whereas update operators (Baral & Zhang 2005). bring a minimal change to each possible world (model) of the base in order to take into account the change described by the new information.

Theorem 12 ([3]). *If $*_\sigma$ is a revision operator (i.e. it satisfies (KB*1)-(KB*6) and (KB*7.3) and Theorem 9 and Lemma 4), then the update operator $\diamond$ defined by $\psi \diamond \mu = \bigvee_{w \models \psi} \psi_{\{w\}} *_\sigma \mu$ is an update operator that satisfies (U1)-(U9).*

7.3 Belief update vs Knowledge base update

Formally speaking, both updates are aiming at maintaining the base of the knowledge or belief up-to-date.

Theorem 13. *If $\circ$ are revision operators (i.e. they satisfy (R1)-(R6)), then the update operator $\diamond$ defined by $\psi \diamond \mu = \bigvee_{w \models \psi} \psi_{\{w\}} \circ \mu$ is an update operator that satisfies (U1)-(U9).*

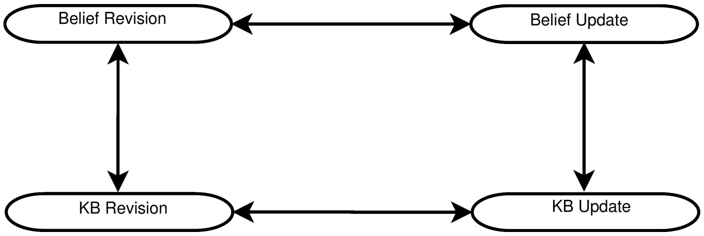


Fig. 2. Belief Update Vs Knowledge base Update

8 Deductive database

A *Deductive database* DDB consists of three parts: an *intensional database* IDB (KB_I), a set of definite program clauses, *extensional database* EDB (KB_U), a set of ground facts; and *integrity constraints* IC. The intuitive meaning of DDB is provided by the *Least Herbrand model semantics* and all the inferences are carried out through SLD-derivation. All the predicates that are defined in IDB are referred to as *view predicates* and those defined in EDB are referred to as *base predicates.* Extending this notion, an atom with a view predicate is said to be a *view atom,* and similarly an atom with base predicate is a *base atom.* Further we assume that IDB does not contain any unit clauses and no predicate defined in a given DDB is both view and base.

Two kinds of view updates can be carried out on a DDB: An atom, that does not currently follow from DDB, can be *inserted,* or an atom, that currently follows from DDB can be *deleted.* When an atom A is to be updated, the view update problem is to insert or delete only some relevant EDB facts, so that the modified EDB together with IDB will satisfy the update of A to DDB.

Note that a DDB can be considered as a Horn knowledge base to be revised. The IDB is the immutable part of the Horn knowledge base dynamics, while the EDB forms the updatable part. In general, it is assumed that the language underlying a DDB is fixed and the semantics of DDB is the least Herbrand model over this fixed language. We assume that there are no function symbols implying that the Herbrand Base is finite. Therefore, the IDB is practically a shorthand of its ground instantiation [3] written as IDB_G. In the sequel, technically we mean IDB_G when we refer simply to IDB. Thus, a DDB represents a Horn knowledge base dynamics where the immutable part is given by IDB_G and updatable part is the EDB. Hence, the rationality postulates (KB*1)-(KB*6) and (KB*7.3) provide an axiomatic characterization for updating (insert and delete) a view atom A into a definite database DDB.

But before discussing the rationality postulates and algorithm, we want to make it precise, how a relational database, along with operations on relations,

[3] a ground instantiation of a definite program P is the set of clauses obtained by substituting terms in the Herbrand Universe for variables in P in all possible ways

can be represented by definite deductive database. We assume the reader is familiar with relational database concepts. *A relation scheme* R can be thought of as a base predicate whose arguments define the *attributes* $\mathbb{A}$ *of the scheme. Its relational extension* r, is a finite set of base atoms $R(\mathbb{A})$ containing the predicate R. A *database schema* consists of finite collection of relational schemes $< R_1, \ldots, R_n >$, and a *relational database* is a specific extension of database schema, denoted as $< r_1, \ldots, r_n >$. In our context, relational database can be represented by $EDB = \bigcup_{i=1,\ldots,n} R_i(\mathbb{A}_i)$.

Join is a binary operator for combining two relations. Let r and s be two relational extensions of schema R (with attributes $\mathbb{R}$) and S (with attributes $\mathbb{S}$), respectively. Let $\mathbb{T} = \mathbb{R} \cup \mathbb{S}$. The join of r and s, written as $r \otimes s$, is the relational extension $q(\mathbb{T})$ of all tuples t over $\mathbb{T}$ such that there are $t_r \in r$ and $t_s \in s$, with $t_r = t(\mathbb{R})$ and $t_s = t(\mathbb{S})$. Join can be captured by a constraint clause $Q(\mathbb{T}) \leftarrow R(\mathbb{R}), S(\mathbb{S})$.

Let us consider two relational schemes R and S from Example 1, with attributes R={Group,Chair} and S={Staff,Group}.Consider the following extensions r and s: (see definition and properties of similarity in works of Christiansen (Christiansen & Rekouts 2007) and Godfrey (Godfrey et al. 1998)).

Example 13.

s	Staff	Group
	delhibabu	infor1
	aravindan	infor2

r	Group	Chair
	infor1	matthias
	infor2	gerhard

Table 1. Base table for s and r

The following rule, T(Staff,Group,Chair) ← S(Staff,Group),R(Group,Chair) represents the join of s and r, which is given in Table 5.2:

$s \otimes r$	$Staff$	$Group$	$Chair$
	delhibabu	infor1	matthias
	aravindan	infor2	gerhard

Table 2. $s \otimes r$

Our first integrity constraint (IC) is that each research group has only one chair ie. $\forall x, y, z$ (y=z) ← group_chair(x,y) ∧ group_chair(x,z). Second integrity constraint is that a person can be a chair for only one research group ie. $\forall x, y, z$ (y=z)← group_chair(y,x) ∧ group_chair(z,x).

An update request U = A, where A is a set of base facts that are not true in KB. Then, we need to find a transaction $T = T_{ins} \cup T_{del}$, where $T_{ins}(\Delta_i)$ (resp.

$T_{del}(\varDelta_j)$) is the set of facts, such that U is true in $DDB' = ((EDB - T_{del} \cup T_{ins}) \cup IDB \cup IC)$. Since we consider stratifiable (definite) deductive databases, SLD-trees can be used to compute the required abductive explanations. The idea is to get all EDB facts used in a SLD-derivation of A with respect to DDB, and construct that as an abductive explanation for A with respect to IDB_G.

Traditional methods translate a view update request into a **transaction combining insertions and deletions of base relations** for satisfying the request (Mota-Herranz et al. 2000). Furthermore, a stratifiable (definite) deductive database can be considered as a knowledge base, and thus the rationality postulates and insertion algorithm from the previous section can be applied for solving view update requests in deductive databases.

There are two ways to find minimal elements (insertion and deletion) in the presence of integrity constraints. Algorithm 2 first checks consistency with integrity constraints and then reduces steps with abductive explanation for A . Algorithm 3 is doing *vice versa*, but both algorithm outputs are similar.

```
Algorithm 2    Algorithm to compute all DDB-closed locally minimal
               abductive explanation of an atom(literals)

Input :   A definite deductive database $DDB = IDB \cup EDB \cup IC$ an literals $\mathcal{A}$
Output :  Set of all DDB-closed locally minimal abductive explanations
          for $\mathcal{A}$ wrt $IDB_G$
begin
  1.  Let $V := \{c \in IC \mid IDB \cup IC$ inconsistent with $\mathcal{A}$ wrt $c$ $\}$
      While $(V \neq 0)$
      Construct a complete SLD-tree for $\leftarrow \mathcal{A}$ wrt DDB.
      For every successful branch $i$: construct $\varDelta_i = \{D \mid D \in EDB$
          and D is used as an input clause in branch $i\}$
      For every unsuccessful branch $j$: construct $\varDelta_j = \{D \mid D \in EDB$
          and D is used as an input clause in branch $j\}$
      Produce set of all $\varDelta_i$ and $\varDelta_j$ computed in the previous step
          as the result.
      return
  2.  Produce all DDB-closed locally minimal abductive
          explanations in $\varDelta_i$ and $\varDelta_j$
end.
```

Horn knowledge base revision algorithm 1, may be applied to compute all DDB-closed locally minimal abductive explanation of an atom (literals). Unfortunately, this algorithm does not work as intended for any deductive database, and a counter example is produced below. Thus, general algorithms 2 and 3 produced some unexpected sets in addition to locally minimal abductive explanations

Algorithm 3 Algorithm to compute all DDB-closed locally minimal abductive explanation of an atom(literals)

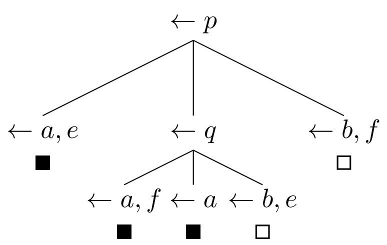
```
Input :   A definite deductive database $DDB = IDB \cup EDB \cup IC$ an literals $\mathcal{A}$
Output :  Set of all DDB-closed locally minimal abductive explanations
          for $\mathcal{A}$ wrt $IDB_G$
begin
  1.  Construct a complete SLD-tree for $\leftarrow \mathcal{A}$ wrt DDB.
      For every successful branch $i$: construct $\Delta_i = \{D \mid D \in EDB$
          and D is used as an input clause in branch $i\}$
      For every unsuccessful branch $j$: construct $\Delta_j = \{D \mid D \in EDB$
          and D is used as an input clause in branch $j\}$
  2.  Let $V := \{c \in IC \mid IDB \cup IC$ inconsistent with $\mathcal{A}$ wrt $c$ $\}$
      While ($V \neq 0$)
          Produce set of all $\Delta_i$ and $\Delta_j$ is consistent with IC
          as the result.
      return
          Produce all DDB-closed locally minimal abductive
          explanations in $\Delta_i$ and $\Delta_j$
end.
```

Example 14. Consider a stratifiable (definite) deductive database DDB as follows:

$$\begin{array}{lll}
IDB : p \leftarrow a \wedge e & EDB : e \leftarrow & IC : \leftarrow b \\
\quad q \leftarrow a \wedge f & \quad f \leftarrow & \\
\quad p \leftarrow b \wedge f & & \\
\quad q \leftarrow b \wedge e & & \\
\quad p \leftarrow q & & \\
\quad q \leftarrow a & &
\end{array}$$

Suppose we want to insert p. First, we need to check consistency with IC and afterwards, we have to find Δ_i and Δ_j via tree deduction.

(Input) $IDB : p \leftarrow a \wedge e, q \leftarrow a \wedge f, p \leftarrow b \wedge f, q \leftarrow b \wedge e, p \leftarrow q, q \leftarrow a$
$EDB : e \leftarrow, f \leftarrow$
$IC : \leftarrow b$
(0) $\{p \leftarrow a, e;\ q \leftarrow a, f;\ p \leftarrow b, f;\ q \leftarrow b, e;\ p \leftarrow q;\ q \leftarrow a;\ e;\ f\}$
(1) $\{V = b\}$
(2)



(3-4) $\Delta_i = \{a, e\}$ and $\Delta_j = \{\}$

(5) $p \leftarrow a, e;\ q \leftarrow a, f;\ p \leftarrow q;\ q \leftarrow a;\ b;\ e;\ f$

(Output) $IDB : p \leftarrow a \wedge e, q \leftarrow a \wedge f, p \leftarrow q, q \leftarrow a$
$EDB' : a \leftarrow, e \leftarrow, f \leftarrow$
$IC : \leftarrow b$

From the step, it is easy to conclude which branches are consistent with respect to IC (indicated in the depicted tree by the symbol ■). For the next step, we need to find minimal accommodate (positive literal) and denial literal (negative literal) with with respect to to p. The subgoals of the tree are $\leftarrow a, e$ and $\leftarrow a, f$, which are minimal tree deductions of only facts. Clearly, $\Delta_i = \{a, e\}$ and $\Delta_j = \{f\}$ with respect to IC, are the only locally minimal abductive explanations for p with respect to IDB_G, but these result are not closed-locally minimal explanations.

For processing a given view update request, a set of all explanations for that atom has to be generated through a complete SLD-tree. The resulting hitting set of these explanations is then a base update of the EDB satsifying the view update request. We present a different approach which is also rational. The generation of a hitting set is carried out through a hyper tableaux calculus (bottom-up) for implementing the deletion process as well as through the magic sets approach (top-down) for performing insertions focussed on the particular goal given.

8.1 View update method

View update (Behrend & Manthey 2008) aims at determining one or more base relation updates such that all given update requests with respect to derived relations are satisfied after the base updates have been successfully applied.

Definition 26 (View update). *Let $DDB = \langle IDB, EDB, IC \rangle$ be a stratifiable (definite) deductive database $DDB(D)$. A VU request ν_D is a pair $\langle \nu_D^+, \nu_D^- \rangle$ where ν_D^+ and ν_D^- are sets of ground atoms representing the facts to be inserted into D or deleted from D, resp., such that $pred(\nu_D^+ \cup \nu_D^-) \subseteq pred(IDB)$, $\nu_D^+ \cap \nu_D^- = \emptyset$, $\nu_D^+ \cap PM_D = \emptyset$ and $\nu_D^- \subseteq PM_D$.*

Note that we consider again true view updates only, i.e. ground atoms which are presently not derivable for atoms to be inserted, or are derivable for atoms to be deleted, respectively. A method for view update determines sets of alternative updates satisfying a given request. A set of updates leaving the given database consistent after its execution is called *VU realization.*

Definition 27 (Induced update). *Let $DDB = \langle IDB, EDB, IC \rangle$ be a stratifiable (definite) deductive database and $DDB = \nu_D$ a VU request. A VU realization is a base update u_D which leads to an induced update $u_{D \to D'}$ from D to D' such that $\nu_D^+ \subseteq PM_{D'}$ and $\nu_D^- \cap PM_{D'} = \emptyset$.*

There may be infinitely many realizations and even realizations of infinite size which satisfy a given VU request. A breadth-first search (BFS) is employed for determining a set of minimal realizations $\tau_D = \{u^1_D, \ldots, u^i_D\}$. Any u^i_D is minimal in the sense that none of its updates can be removed without losing the property of being a realization for ν_D.

Magic Set (Top-down computation):

Given a VU request ν_{DDB}, view update methods usually determine further VU requests in order to find relevant base updates. Similar to delta relations for UP we will use the notion VU relation to access individual view updates with respect to the relations of our system. For each relation $p \in pred(IDB \cup EDB)$ we use the VU relation $\nabla^+_p(\boldsymbol{x})$ for tuples to be inserted into DDB and $\nabla^-_p(\boldsymbol{x})$ for tuples to be deleted from DDB. The initial set of delta facts resulting from a given VU request is again represented by so-called *VU seeds*.

Definition 28 (View update seeds). *Let $DDB(D)$ be a stratifiable (definite) deductive database and $\nu_{DDB} = \langle \nu^+_D, \nu^-_D \rangle$ a VU request. The set of VU seeds $vu\_seeds(\nu_D)$ with respect to ν_D is defined as follows:*

$$vu\_seeds(\nu_D) := \left\{ \nabla^\pi_p(c_1, \ldots, c_n) | p(c_1, \ldots, c_n) \in \nu^\pi_D \text{ and } \pi \in \{+, -\} \right\}.$$

Definition 29 (View update rules). *Let IDB be a normalized stratifiable (definite) deductive rule set. The set of VU rules for true view updates is denoted IDB^∇ and is defined as the smallest set satisfying the following conditions:*

1. *For each rule of the form $p(\boldsymbol{x}) \leftarrow q(\boldsymbol{y}) \wedge r(\boldsymbol{z}) \in IDB$ with $vars(p(\boldsymbol{x})) = (vars(q(\boldsymbol{y})) \cup vars(r(\boldsymbol{z})))$ the following three VU rules are in IDB^∇:*

$$\begin{array}{ll} \nabla^+_p(\boldsymbol{x}) \wedge \neg q(\boldsymbol{y}) \rightarrow \nabla^+_q(\boldsymbol{y}) & \nabla^-_p(\boldsymbol{x}) \rightarrow \nabla^-_q(\boldsymbol{y}) \vee \nabla^-_r(\boldsymbol{z}) \\ \nabla^+_p(\boldsymbol{x}) \wedge \neg r(\boldsymbol{z}) \rightarrow \nabla^+_r(\boldsymbol{z}) & \end{array}$$

2. *For each rule of the form $p(\boldsymbol{x}) \leftarrow q(\boldsymbol{x}) \wedge \neg r(\boldsymbol{x}) \in IDB$ the following three VU rules are in IDB^∇:*

$$\begin{array}{ll} \nabla^+_p(\boldsymbol{x}) \wedge \neg q(\boldsymbol{x}) \rightarrow \nabla^+_q(\boldsymbol{x}) & \nabla^-_p(\boldsymbol{x}) \rightarrow \nabla^-_q(\boldsymbol{x}) \vee \nabla^+_r(\boldsymbol{x}) \\ \nabla^+_p(\boldsymbol{x}) \wedge r(\boldsymbol{x}) \rightarrow \nabla^-_r(\boldsymbol{x}) & \end{array}$$

3. *For each two rules of the form $p(\boldsymbol{x}) \leftarrow q(\boldsymbol{x})$ and $p(\boldsymbol{x}) \leftarrow r(\boldsymbol{x})$ the following three VU rules are in IDB^∇:*

$$\begin{array}{ll} \nabla^-_p(\boldsymbol{x}) \wedge q(\boldsymbol{x}) \rightarrow \nabla^-_q(\boldsymbol{x}) & \nabla^+_p(\boldsymbol{x}) \rightarrow \nabla^+_q(\boldsymbol{x}) \vee \nabla^+_r(\boldsymbol{x}) \\ \nabla^-_p(\boldsymbol{x}) \wedge r(\boldsymbol{x}) \rightarrow \nabla^-_r(\boldsymbol{x}) & \end{array}$$

4. a) *For each relation p defined by a single rule $p(\boldsymbol{x}) \leftarrow q(\boldsymbol{y}) \in IDB$ with $vars(p(\boldsymbol{x})) = vars(q(\boldsymbol{y}))$ the following two VU rules are in IDB^∇:*

$$\nabla^+_p(\boldsymbol{x}) \rightarrow \nabla^+_q(\boldsymbol{y}) \quad \nabla^-_p(\boldsymbol{x}) \rightarrow \nabla^-_q(\boldsymbol{y})$$

b) For each relation p defined by a single rule $p \leftarrow \neg q \in IDB$ the following two VU rules are in IDB^{∇}:

$$\nabla_p^+ \rightarrow \nabla_q^- \quad \nabla_p^- \rightarrow \nabla_q^+$$

5. Assume without loss of generality that each projection rule in IDB is of the form $p(\boldsymbol{x}) \leftarrow q(\boldsymbol{x}, Y) \in IDB$ with $Y \notin vars(p(\boldsymbol{x}))$. Then the following two VU rules

$$\nabla^{-}p_{(}\boldsymbol{x}) \wedge q(\boldsymbol{x}, Y) \rightarrow \nabla_q^-(\boldsymbol{x}, Y)$$
$$\nabla_p^+(\boldsymbol{x}) \rightarrow \nabla_q^+(\boldsymbol{x}, c_1) \vee \ldots \vee \nabla_q^+(\boldsymbol{x}, c_n) \vee \nabla_q^+(\boldsymbol{x}, c^{new})$$

are in IDB^{∇} where all c_i are constants from the Herbrand universe $\mathcal{U}_{DDB}$ of DDB and c^{new} is a new constant, i.e. $c^{new} \notin \mathcal{U}_{DDB}$.

Theorem 14. *Let $DDB = \langle IDB, EDB, IC \rangle$ be a stratifiable (definite)deductive database(D), ν_D a view update request and $\tau_D = \{u_D^1, \ldots, u_D^n\}$ the corresponding set of minimal realizations. Let $D^{\nabla} = \langle EDB \cup vu\_seeds(\nu_D), IDB \cup IDB^{\nabla} \rangle$ be the transformed deductive database of D. Then the VU relations in PM_D^{∇} with respect to base relations of D correctly represent all direct consequences of ν_D. That is, for each realization $u_D^i = \langle u_D^{i^+}, u_D^{i^-} \rangle \in \tau_D$ the following condition holds:*

$$\exists p(\boldsymbol{t}) \in u_D^{i^+} : \nabla_p^+(\boldsymbol{t}) \in MS_D^{\nabla} \vee \exists p(\boldsymbol{t}) \in u_D^{i^-} : \nabla_p^-(\boldsymbol{t}) \in MS_D^{\nabla}.$$

Proof. Follows from the result of (Behrend & Manthey 2008).

Hyper Tableau (Bottom-up computation):

In (Aravindan & Baumgartner 1997) a variant of clausal normal form tableaux called "hyper tableaux" is introduced for view deletion method. Since the hyper tableaux calculus constitutes the basis for our view update algorithm, *Clauses,* i.e. multisets of literals, are usually written as the disjunction $A_1 \vee A_2 \vee \cdots \vee A_m \vee$ not $B_1 \vee$ not B_2
$\cdots \vee$ not B_n $(M \geq 0, n \geq 0)$. The literals $A_1, A_2, \ldots A_m$ (resp. $B_1, B_2, \ldots, B_n$) are called the *head (*resp. *body)* of a clause. With $\overline{L}$ we denote the complement of a literal L. Two literals L and K are complementary if $\overline{L} = K$.

From now on D always denotes a finite ground clause set, also called *database,* and Σ denotes its signature, i.e. the set of all predicate symbols occurring in it. We consider finite ordered trees T where the nodes, except the root node, are labeled with literals. In the following we will represent a branch b in T by the sequence $b = L_1, L_2, \ldots, L_n$ $(n \geq 0)$ of its literal labels, where L_1 labels an immediate successor of the root node, and L_n labels the leaf of b. The branch b is called *regular* iff $L_i \neq L_j$ for $1 \leq i, j \leq n$ and $i \neq j$, otherwise it is called *irregular.* The tree T is *regular* iff every of its branches is regular, otherwise it is *irregular.* The set of *branch literals* of b is $lit(b) = \{L_1, L_2, \ldots, L_n\}$. For brevity, we will write expressions like $A \in b$ instead of $A \in lit(b)$. In order to memorize the fact that a branch contains a contradiction, we allow to label a branch as either *open* or *closed.* A tableau is closed if each of its branches is closed, otherwise it is open.

Definition 30 (Hyper Tableau). *A literal set is called inconsistent* iff it contains a pair of complementary literals, otherwise it is called *consistent. Hyper tableaux* for D are inductively defined as follows:

Initialization step: The empty tree, consisting of the root node only, is a hyper tableau for D. Its single branch is marked as "open".

Hyper extension step: If (1) T is an open hyper tableau for D with open branch b, and (2) $C = A_1 \vee A_2 \vee \cdots \vee A_m \leftarrow B_1 \wedge B_2 \cdots \wedge B_n$ is a clause from D ($n \geq 0, m \geq 0$), called *extending clause* in this context, and (3) $\{B_1, B_2, \ldots, B_n\} \subseteq b$ (equivalently, we say that C is *applicable to b)*then the tree T is a hyper tableau for D, where T is obtained from T by extension of b by C: replace b in T by the *new branches*

$$(b, A_1), (b, A_2), \ldots, (b, A_m), (b, \neg B_1), (b, \neg B_2), \ldots, (b, \neg B_n)$$

and then mark every inconsistent new branch as "closed", and the other new branches as "open".

The applicability condition of an extension expresses that all body literals have to be satisfied by the branch to be extended. From now on, we consider only regular hyper tableaux. This restriction guarantees that for finite clause sets no branch can be extended infinitely often. Hence, in particular, no open finished branch can be extended any further. This fact will be made use of below occasionally. Notice as an immediate consequence of the above definition that open branches never contain negative literals.

This paper work focused on stratified (definite) deductive database without any auxiliary variable. In magic set rule play in minimal case, our future goal is similar foundation using auxiliary variable (Deductive Databases) side and more details found in Behrend's (Behrend & Manthey 2008) work.

8.2 View update algorithm

The key idea of the algorithm presented in this paper is to transform the given database along with the view update request into a logic program and apply known minimality techniques to solve the original view update problem. The intuition behind the transformation is to obtain a logic program in such a way that each (minimal) model of this transformed program represent a way to update the given view atom. We present two variants of our algorithm. The one that is discussed in this section employs a trivial transformation procedure but has to look for minimal models; and another performs a costly transformation, but dispenses with the requirement of computing the minimal models.

Minimality test

We start presenting an algorithm for stratifiable (definite) deductive databases by first defining precisely how the given database is transformed into a logic program for the view deletion process (Aravindan & Baumgartner 1997)

Definition 31 **(IDB Transformation).** *Given an IDB and a set of ground atoms S, the transformation of IDB with respect to S is obtained by translating each clause $C \in IDB$ as follows: Every atom A in the body (resp. head) of C that is also in S is moved to the head (resp. body) as $\neg A$.*

Note 5. If IDB is a stratifiable deductive database then the transformation introduced above is not necessary.

Definition 32 **(IDB^* Transformation).** *Let $IDB \cup EDB$ be a given database. Let $S_0 = EDB \cup \{A \mid A$ is a ground IDB* atom$\}$*. Then, IDB^* is defined as the transformation of IDB with respect to S_0.*

Note 6. IDB^* is in general a logic program. The negative literals ($\neg A$) appearing in the clauses are intuitively interpreted as deletion of the corresponding atom (A) from the database. Technically, a literal $\neg A$ is to be read as a *positive* atom, by taking the $\neg$-sign as part of the predicate symbol. To be more precise, we treat $\neg A$ as an atom with respect to IDB^*, but as a negative literal with respect to IDB.

Note that there are no facts in IDB^*. So when we add a delete request such as $\neg A$ to this, the added request is the only fact and any bottom-up reasoning strategy is fully focused on the goal (here the delete request)

Definition 33 (Update Tableaux Hitting Set). *An update tableau for a database $IDB \cup EDB$ and delete request $\neg A$ is a hyper tableau T for $IDB^* \cup \{\neg A \leftarrow\}$ such that every open branch is finished. For every open finished branch b in T we define the hitting set (of b in T)* as $HS(b) = \{A \in EDB | \neg A \in b\}$.

The next step is to consider the view insertion process (Behrend & Manthey 2008):

Definition 34 **($IDB^\bullet$ Transformation).** *Let $IDB \cup EDB$ be a given database. Let $S_1 = EDB \cup \{A \mid A$ is a ground IDB* atom$\}$ *(that is either body or head empty). Then, $IDB^\bullet$ is defined as the transformation of IDB with respect to S_1.*

Note 7. IDB is in general a (stratifiable) logic program. The positive literals (A) appearing in the clauses are intuitively interpreted as an insertion of the corresponding atom (A) from the database.

Definition 35 (Update magic Hitting Set). *An update magic set rule for a database $IDB \cup EDB$ and insertion request A is a magic set rule M for $IDB^\bullet \cup \{A \leftarrow\}$ such that every close branch is finished. For every close finished branch b in M we define the magic set rule (of b in M)* as $HS(b) = \{A \in EDB | A \in b\}$.

Example 15. Given stratifiable (definite) deductive database $DDB = IDB \cup EDB \cup IC$ and insert p.

$$\begin{array}{rlll} IDB: & p \leftarrow a \wedge e & EDB: a \leftarrow & IC: \emptyset \\ & q \leftarrow a \wedge e & c \leftarrow & \\ & p \leftarrow a \wedge f & & \\ & q \leftarrow c & & \end{array}$$

IDB^* Transformation:

$$\begin{array}{rlll} IDB^*: & \neg a \vee \neg e \leftarrow \neg p & EDB: a \leftarrow & IC: \emptyset \\ & \neg a \vee \neg e \leftarrow \neg q & c \leftarrow & \\ & \neg a \vee \neg f \leftarrow \neg p & & \\ & \neg c \leftarrow \neg q & & \end{array}$$

$IDB^\bullet$ Transformation: (Body empty)

$$\begin{array}{rlll} IDB^\bullet: & p \vee \neg a \vee \neg e \leftarrow & EDB: a \leftarrow & IC: \emptyset \\ & q \vee \neg a \vee \neg e \leftarrow & c \leftarrow & \\ & p \vee \neg a \vee \neg f \leftarrow & & \\ & q \vee \neg c \leftarrow & & \end{array}$$

$IDB^\bullet$ Transformation: (Head empty)

$$\begin{array}{rlll} IDB^\bullet: & \leftarrow \neg p \wedge a \wedge e & EDB: a \leftarrow & IC: \emptyset \\ & \leftarrow \neg q \wedge a \wedge e & c \leftarrow & \\ & \leftarrow \neg p \wedge a \wedge f & & \\ & \leftarrow \neg q \wedge c & & \end{array}$$

The set S_0 is determined by all the IDB atoms and the current EDB atoms and in our case it is $\{p, q, a, c, e, f\}$. IDB^* and $IDB^\bullet$ is the transformation of IDB with respect to S_0 which is given above.

Suppose a ground view atom $\mathcal{A}$ is to be insert. Then, an update tableau for $IDB^\bullet$ with insert request $\mathcal{A}$ (IDB^* with delete request $\neg\mathcal{A}$) is built. The intuition is that the set of EDB atoms appearing in a model (open/close branch) constitute a hitting set, and removing/adding this set from EDB should achieve the required view insertion. Unfortunately, this does not result in a rational insertion, as relevance policy may be violated.

Example 16. Let us continue with example 15 Suppose the view atom p is to be insert. Then according to the above proposal, an update tableau for $IDB^\bullet$ (IDB^*) and p ($\neg p$) is to be built. This is illustrated in the accompanying figure below. As shown, open/close branches constitute two hitting sets $\{a\}$ and $\{f, a\}$ ($\{\neg a\}$ and $\{\neg f, \neg a\}$). It is not difficult to see that $\{f, a\}(\{\neg f, \neg a\})$ does not satisfy any of the relevance policies (KB*7.1) or (KB*7.2) or (KB*7.3). Hence simple model computation using hyper tableau calculus does not result in rational hitting sets. The branch is closed if the corresponding hitting set does not satisfy this strong relevance postulate.

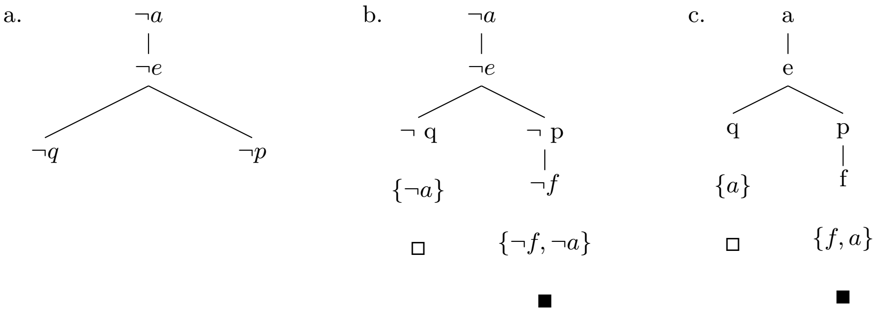


Fig. 3. IDB^* and $IDB^\bullet$ transformation with hitting set

Definition 36 (Minimality test). *Let T be an update tableau for $IDB^* \cup EDB$ and delete request $\neg A$. We say that open finished branch b in T satisfies the strong minimality test* iff $\forall s \in HS(b) : IDB^* \cup EDB \backslash HS(b) \cup \{s\} \vdash \neg A$.

Definition 37 (Update Tableau satisfying strong minimality). *An update tableau for given $IDB \cup EDB$ and delete request $\neg A$ is transformed into an update tableau satisfying strong minimality by marking every open finished branch as closed which does not satisfy strong minimality.*

The next step is to consider the view insertion process (Behrend & Manthey 2008):

Definition 38 (Minimality test). *Let M be an update magic set rule for $IDB \cup EDB$ and insert request A. We say that close finished branch b in M satisfies the strong minimality test* iff $\forall s \in HS(b) : IDB^\bullet \cup EDB \backslash HS(b) \cup \{s\} \vdash A$.

Definition 39 (Update magic set rule satisfying strong minimality). *An update magic set rule for given $IDB \cup EDB$ and insert request A is transformed into an update magic set rule satisfying strong minimality by marking every close finished branch as open which does not satisfy strong minimality.*

Example 17. Continuing with the above example, after constructing the branch corresponding to the hitting set $\{f, a\}(\{\neg f, \neg a\})$, the strong minimality test is carried out as follows: It is checked if the resulting database with each member of hitting set implies the insert atom p. For example, $IDB \cup EDB \backslash \{f, a\} \cup \{a\} \nvdash p$, and hence this branch fails the strong minimality test.

Interestingly, this minimality test is equivalent to the groundedness test used for generating minimal models of logic programs. The key idea of the groundedness test is to check if the members in the model are implied by the program together with the negation/positive of the atoms not present in the model. The groundedness test for generating minimal models can be stated as follows: Let T be an update tableau for $IDB \cup EDB$ and insert request $\mathcal{A}$.We say that open finished branch b in T satisfies the groundedness test iff $\forall s \in HS(b) : IDB^{\bullet} \cup EDB \backslash HS(b) \cup \{\mathcal{A}\} \vdash s$, similar for IDB^{*} $(\forall s \in HS(b) : IDB^{*} \cup EDB \backslash HS(b) \cup \{\neg\mathcal{A}\} \vdash \neg s)$. It is not difficult to see that this is equivalent to the minimality test. This means that every minimal model (minimal with respect to the base atoms) of $IDB^{*} \cup \{\mathcal{A}\}$ provides a minimal hitting set for insertion the ground view atom $\mathcal{A}$.

Algorithm 4 View update Algorithm based on minimality test

```
Input :   A stratifiable (definite) deductive database $DDB = IDB \cup EDB \cup IC$
          an literals $\mathcal{A}$
Output:   A new stratifiable (definite) database $IDB \cup EDB' \cup IC$
begin
  1.  Let $V := \{c \in IC \mid IDB \cup IC$ inconsistent with $\mathcal{A}$ with respect to $c$ $\}$
      While $(V \neq \emptyset)$
  2.  Construct a complete SLD-tree for $\leftarrow \mathcal{A}$ with respect to DDB.
  3.  For every successful branch $i$:construct $\Delta_i = \{D \mid D \in EDB\}$
      and D is used as an input clause in branch $i$.
      Construct a branch i of an update tableau satisfying minimality
      for $IDB \cup EDB$ and delete request $\neg A$.
      Produce $IDB \cup EDB \backslash HS(i)$ as a result
  4.  For every unsuccessful branch $j$:construct $\Delta_j = \{D \mid D \in EDB\}$
      and D is used as an input clause in branch $j$.
      Construct a branch j of an update magic set rule satisfying minimality
      for $IDB \cup EDB$ and insert request $A$.
      Produce $IDB \cup EDB \backslash HS(j)$ as a result
      Let $V := \{c \in IC \mid IDB \cup IC$ inconsistent with $\mathcal{A}$ with respect to $c$ $\}$
      return
  5.  Produce $DDB$ as the result.
end.
```

This means that every minimal model (minimal with respect to the base atoms) of $IDB^{*} \cup \{\neg A\}$ $(IDB^{\bullet} \cup \{A\}$)provides a minimal hitting set for deleting the ground view atom A. Similarly, $IDB^{*} \cup \{A\}$ provides a minimal hitting set for inserting the ground view atom A. We are formally present our algorithms.

Given a database and a view atom to be updated, we first transform the database into a logic program and use hyper tableaux calculus to generate models of this transformed program for deletion of an atom. Second, magic sets transformed rules are used is used to generate models of this transformed program for determining an induced insertion of an atom. Models that do not represent rational update are filtered out using the strong minimality test. The procedure for stratifiable (definite) deductive databases is presented in Algorithms in 4 and 5.

```
Algorithm 5        View update Algorithm based on minimality test
Input :    A stratifiable (definite) deductive database $DDB = IDB \cup EDB \cup IC$
           an literals $\mathcal{A}$
Output:    A new stratifiable (definite) database $IDB \cup EDB' \cup IC$
begin
  1.  Construct a complete SLD-tree for $\leftarrow \mathcal{A}$ with respect to DDB.
  2.  For every successful branch $i$:construct $\Delta_i = \{D \mid D \in EDB\}$
          and D is used as an input clause in branch $i$.
  3.  For every unsuccessful branch $j$:construct $\Delta_j = \{D \mid D \in EDB\}$
          and D is used as an input clause in branch $j$.
  4.  Let $V := \{c \in IC \mid IDB \cup IC$ inconsistent with $\mathcal{A}$ with respect to $c$ $\}$
      While $(V \neq \emptyset)$
          Construct a branch i of an update tableau satisfying minimality
              for $IDB \cup EDB$ and delete request $\neg A$.
          Produce $IDB \cup EDB \backslash HS(i)$ as a result
          Construct a branch j of an update magic set rule satisfying minimality
              for $IDB \cup EDB$ and insert request $A$.
          Produce $IDB \cup EDB \backslash HS(j)$ as a result
          Let $V := \{c \in IC \mid IDB \cup IC$ inconsistent with $\mathcal{A}$ with respect to $c$ $\}$
      return
  5.  Produce $DDB$ as the result.
end.
```

Lemma 1. *The strong minimality test and the groundedness test are equivalent.*

Proof. Follows from the result of (Aravindan & Baumgartner 1997).

Example 18.

$$\begin{array}{rlrll}
IDB: & p \leftarrow a \wedge e & EDB: & a \leftarrow & IC: \leftarrow b \\
& q \leftarrow a \wedge e & & f \leftarrow & \\
& p \leftarrow b \wedge f & & & \\
& q \leftarrow b \wedge f & & & \\
& p \leftarrow g \wedge a & & & \\
& q \leftarrow p & & &
\end{array}$$

Suppose we want to insert p. First, we need to check consistency with IC and afterwards, we have to find Δ_i and Δ_j via tree deduction.

From algorithm 4 or 5 (only different is checking IC condition), the above example execute following steps:

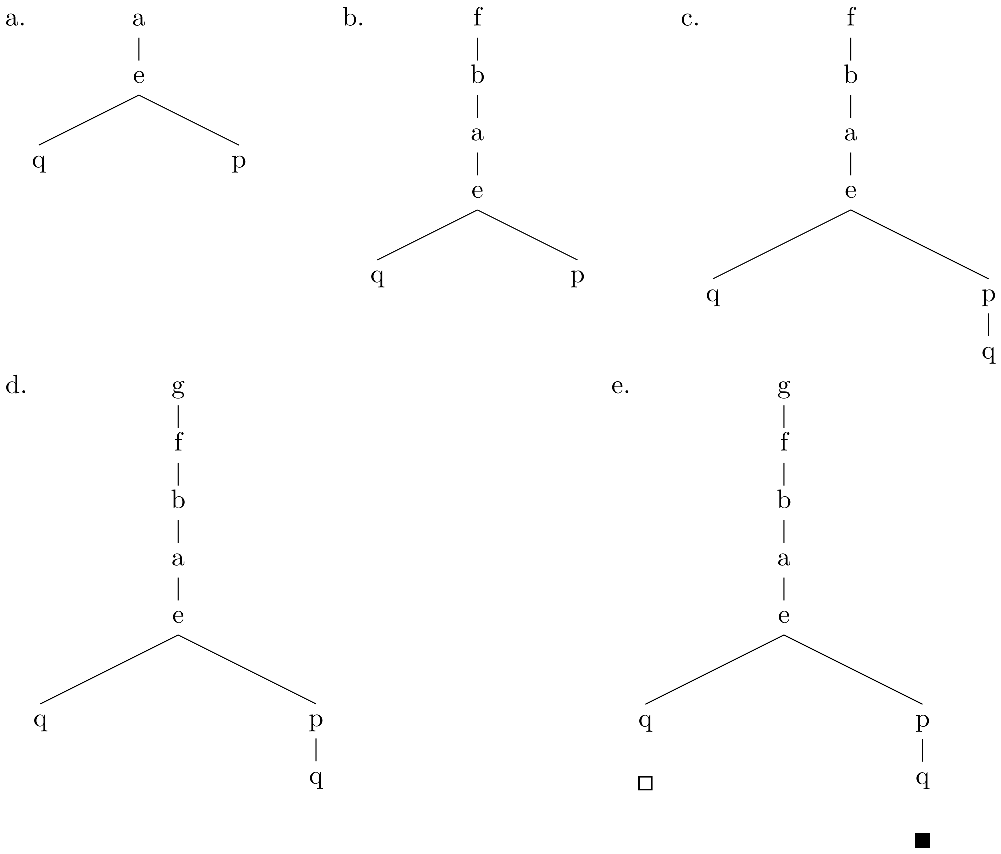
Step number with execution

(Input) $IDB: p \leftarrow a \wedge e;\ q \leftarrow a \wedge e;\ p \leftarrow b \wedge f;\ q \leftarrow b \wedge f;\ p \leftarrow g \wedge a;\ q \leftarrow p$
$EDB: a \leftarrow, f \leftarrow$
$IC: \leftarrow b$

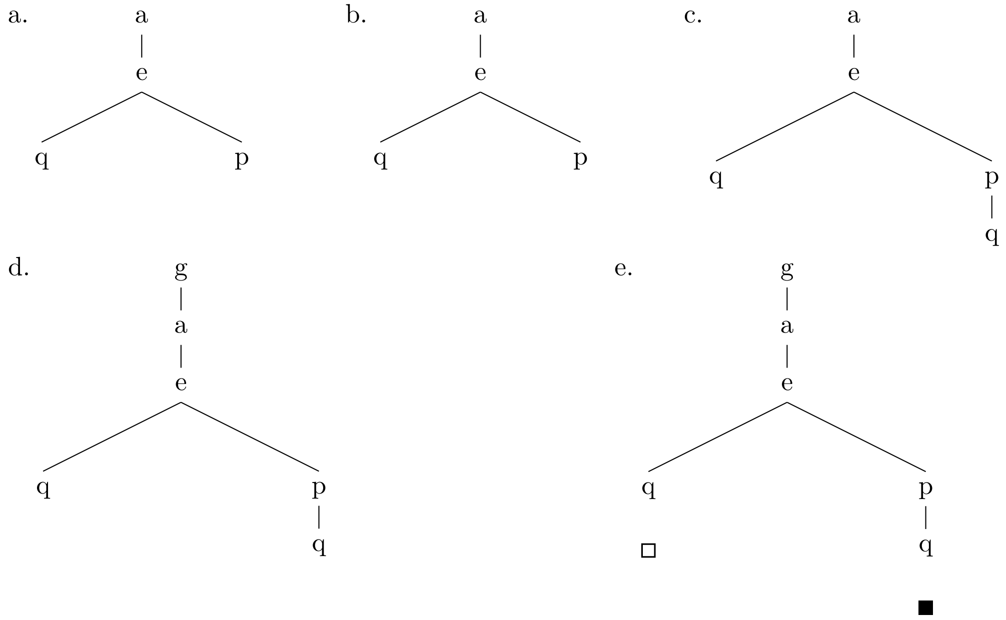
(0) $\{p \leftarrow a, e;\ q \leftarrow a, e;\ p \leftarrow b, f;\ q \leftarrow b, f;\ p \leftarrow g, a;\ q \leftarrow p;\ a;\ f\}$

(1) $\{V = b\}$

(2.1)



(2.2)



(3-4) $\Delta_i = \{a, e, g\}$ and $\Delta_j = \{\}$
(5) $p \leftarrow a, e;\ q \leftarrow a, e;\ p \leftarrow g, a;\ q \leftarrow p;\ a;\ e;\ f;\ g$
(Output) $IDB : p \leftarrow a \wedge e;\ q \leftarrow a \wedge e;\ p \leftarrow g \wedge a;\ q \leftarrow p$
$EDB' : a \leftarrow, e \leftarrow, f \leftarrow, g \leftarrow$
$IC : \leftarrow b$

To show the rationality of this approach, we study how this is related to the previous approach presented in the last section, i.e. generating explanations and computing hitting sets of these explanations. To better understand the relationship it is imperative to study where the explanations are in the hyper tableau approach and magic set rules. We first define the notion of an *EDB*-cut and then view update seeds.

Definition 40 **(*EDB*-Cut).** *Let T be update tableau with open branches $b_1, b_2, \ldots, b_n$. A set $S = \{A_1, A_2, \ldots, A_n\} \subseteq EDB$ is said to be EDB-cut of T iff $\neg A_i \in b_i$ ($A_i \in b_i$), for $1 \leq i \leq n$.*

Definition 41 **(*EDB* seeds).** *Let M be an update seeds with close branches $b_1, b_2, \ldots, b_n$. A set $S = \{A_1, A_2, \ldots, A_n\} \subseteq EDB$ is said to be a EDB-seeds of M iff EDB seeds vu_seeds(ν_D) with respect to ν_D is defined as follows:*

$$vu\_seeds(\nu_D) := \left\{ \nabla_p^{\pi}(c_1, \ldots, c_n) | p(c_1, \ldots, c_n) \in \nu_D^{\pi} \text{ and } \pi \in \{+, -\} \right\}.$$

Lemma 2. *Let T be an update tableau for $IDB \cup EDB$ and update request A. Similarly, for M be an update magic set rule. Let S be the set of all EDB-closed minimal abductive explanations for A with respect to. IDB. Let S' be the set of all EDB-cuts of T and EDB-seeds of M . Then the following hold*

- $S \subseteq S'$.
- $\forall \Delta' \in S' : \exists \Delta \in S s.t. \Delta \subseteq \Delta'$.

Proof.

1. Consider a $\Delta(\Delta \in \Delta_i \cup \Delta_j) \in S$. We need to show that Δ is generated by algorithm 4 at step 2. From observation 4, it is clear that there exists a A-kernel X of DDB_G s.t. $X \cap EDB = \Delta_j$ and $X \cup EDB = \Delta_i$. Since $X \vdash A$, there must exist a successful derivation for A using only the elements of X as input clauses and similarly $X \nvdash A$. Consequently Δ must have been constructed at step 2.
2. Consider a $\Delta'((\Delta' \in \Delta_i \cup \Delta_j) \in S'$. Let Δ' be constructed from a successful (unsuccessful) branch i via $\Delta_i(\Delta_j)$. Let X be the set of all input clauses used in the refutation i. Clearly $X \vdash A(X \nvdash A)$. Further, there exists a minimal (with respect to set-inclusion) subset Y of X that derives A (i.e. no proper subset of Y derives A). Let $\Delta = Y \cap EDB$ $(Y \cup EDB)$. Since IDB does not (does) have any unit clauses, Y must contain some EDB facts, and so Δ is not empty (empty) and obviously $\Delta \subseteq \Delta'$. But, Y need not (need) be a A-kernel for IDB_G since Y is not ground in general. But it stands for several A-kernels with the same (different) EDB facts Δ in them. Thus, from observation 4, Δ is a DDB-closed locally minimal abductive explanation for A with respect to IDB_G and is contained in Δ'. minimal.

The above lemma precisely characterizes what explanations are generated by an update tableau. It is obvious then that a branch cuts through all the explanations and constitutes a hitting set for all the generated explanations. This is formalized below.

Lemma 3. *Let S and S' be sets of sets s.t. $S \subseteq S'$ and every member of $S' \backslash S$ contains an element of S. Then, a set H is a minimal hitting set for S iff it is a minimal hitting set for S'.*

Proof.

1. (**Only if part**) Suppose H is a minimal hitting set for S. Since $S \subseteq S'$, it follows that $H \subseteq \bigcup S'$. Further, H hits every element of S', which is evident from the fact that every element of S' contains an element of S. Hence H is a hitting set for S'. By the same arguments, it is not difficult to see that H is minimal for S' too.

   (**If part**) Given that H is a minimal hitting set for S', we have to show that it is a minimal hitting set for S too. Assume that there is an element $E \in H$ that is not in $\bigcup S$. This means that E is selected from some $Y \in S' \backslash S$. But Y contains an element of S, say X. Since X is also a member of S', one member of X must appear in H. This implies that two elements have been selected from Y and hence H is not minimal. This is a contradiction and hence $H \subseteq \bigcup S$. Since $S \subseteq S'$, it is clear that H hits every element in S, and so H is a hitting set for S. It remains to be shown that H is minimal. Assume the contrary, that a proper subset H' of H is a hitting set for S. Then from the proof of the only if part, it follows that H' is a hitting set for S' too, and contradicts the fact that H is a minimal hitting set for S' . Hence, H must be a minimal hitting set for S.

Lemma 4. *Let T be an update tableau for $IDB \cup EDB$ and update request A that satisfies the strong minimality test. Similarly, for M be an updating magic set rule. Then, for every open (close) finished branch b in T, $HS(b)$ (M, $HS(b)$) is a minimal hitting set of all the abductive explanations of A.*

Proof. Follows from the Observation 4 (minimal test) in and (Behrend & Manthey 2008).

So, Algorithms 4 and 5 generate a minimal hitting set (in polynomial space) of all EDB-closed locally minimal abductive explanations of the view atom to be deleted. From the belief dynamics results recalled in section 5.3, it immediately follows that Algorithms 4 and 5 are rational, and satisfy the strong relevance postulate (KB*7.1).

Theorem 15. *Algorithms 4 and 5 are rational, in the sense that they satisfy all the rationality postulates (KB*1)-(KB*6) and the strong relevance postulate (KB*7.1). Further, any update that satisfies these postulates can be computed by these algorithms.*

Proof. Follows from Observation 4,4 and Theorem 7.

8.3 Materialized view

In many cases, the view update to be materialized, i.e. the least Herbrand Model is computed and kept, for efficient query answering. In such a situation, rational hitting sets can be computed without performing any minimality test. The idea is to transform the given IDB with respect to the materialized view.

Definition 42 (IDB^+ Transformation). *Let $IDB \cup EDB$ be a given database. Let S be the Least Herbrand Model of this database. Then, IDB^+ is defined as the transformation of IDB with respect to S.*

Note 8. If IDB is a stratifiable deductive database then the transformation introduced above is not necessary.

Definition 43 (Update Tableau based on Materialized view). *An update tableau based on materialized view for a database $IDB \cup EDB$ and delete request $\neg A$ is a hyper tableau T for $IDB^+ \cup \{\neg A \leftarrow\}$ such that every open branch is finished.*

Definition 44 (IDB^- Transformation). *Let $IDB \cup EDB$ be a given database. Let S_1 be the Least Herbrand Model of this database (that is either body or head empty). Then, IDB^- is defined as the transformation of IDB with respect to S_1.*

Definition 45 (Update magic set rule based on Materialized view). *An update magic set rule based on materialized view for a database $IDB \cup EDB$ and insert request A is a magic set M for $IDB^+ \cup \{A \leftarrow\}$ such that every close branch is finished.*

Now the claim is that every model of $IDB^+ \cup \{\neg A \leftarrow\}$ $(A \leftarrow)$ constitutes a rational hitting set for the deletion and insertion of the ground view atom A. So, the algorithm works as follows: Given a database and a view update request, we first transform the database with respect to its Least Herbrand Model (computation of the Least Herbrand Model can be done as a offline preprocessing step. Note that it serves as materialized view for efficient query answering). Then the hyper tableaux calculus (magic set rule) is used to compute models of this transformed program. Each model represents a rational way of accomplishing the given view update request. This is formalized in Algorithms 6 and 7.

Like the approach with minimality test, this algorithm runs on not polynomial space. This approach require minimality test, but our focus on integrity constrain open/close branch. Again, this requires some offline pre-processing of computing the Least Herbrand Model. Note that, our future direction to construct minimality test, this method may generate a non-minimal (but rational) hitting set.

Example 19. Given stratifiable (definite) deductive database $DDB = IDB \cup EDB \cup IC$ and insert p.

$$
\begin{array}{rcrcl}
IDB: & p \leftarrow a & EDB: c \leftarrow & & IC: \emptyset \\
 & q \leftarrow a & d \leftarrow & & \\
 & q \leftarrow c \wedge b & & & \\
 & q \leftarrow p & & &
\end{array}
$$

IDB^+ Transformation:

$$
\begin{array}{rcrcl}
IDB^+: & \neg a \leftarrow \neg p & EDB: c \leftarrow & & IC: \emptyset \\
 & \neg a \leftarrow \neg q & d \leftarrow & & \\
 & \neg c \vee \neg b \leftarrow \neg p & & & \\
 & \neg p \leftarrow \neg q & & &
\end{array}
$$

IDB^- Transformation: (Body empty)

$$
\begin{array}{rcrcl}
IDB^-: & p \vee \neg a \leftarrow & EDB: c \leftarrow & & IC: \emptyset \\
 & q \vee \neg a \leftarrow & d \leftarrow & & \\
 & q \vee \neg c \vee \neg b \leftarrow & & & \\
 & q \vee \neg p \leftarrow & & &
\end{array}
$$

IDB^- Transformation: (Head empty)

$$\begin{array}{llll} IDB^- : & \leftarrow \neg p \wedge a & EDB : c \leftarrow & IC : \emptyset \\ & \leftarrow \neg q \wedge a & \quad d \leftarrow & \\ & \leftarrow \neg g \wedge c \wedge b & & \\ & \leftarrow \neg q \wedge p & & \end{array}$$

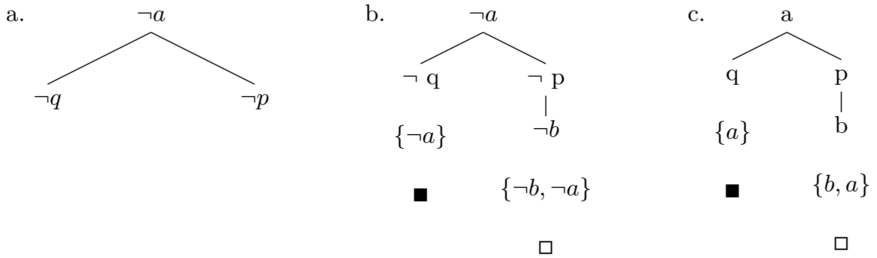


Fig. 4. IDB^+ and IDB^- transformation with hitting set

The Least Herbrand Model of this database is $\{p, q, a, b\}$. The transformed database IDB^+ and IDB^- based on this model, together with an update tableaux for insertion request p based on materialised view is as above figure:

Observe that the last two clauses are never used and the necessarily failing attempt of deleting t to delete p is never made, thus greatly reducing the search space. Also note that the two cuts with only EDB atoms $\{a, b\}$ and $\{a\}$ are exactly the two locally minimal explanations for p. The two open branches provide the two models of $IDB^+ \cup \{\neg p\}$ $(IDB^- \cup \{p\}$ which stand for the hitting sets $\{a, b\}$ and $\{a\}$. Clearly, $\{a, b\}$ not minimal.

```
Algorithm 6        View update algorithm based on Materialized view
Input :    A stratifiable (definite) deductive database $DDB = IDB \cup EDB \cup IC$
           an literals $\mathcal{A}$
Output:    A new stratifiable (definite) database $IDB \cup EDB' \cup IC$
begin
1.    Let $V := \{c \in IC \mid IDB \cup IC$ inconsistent with $\mathcal{A}$ with respect to $c$ $\}$
      While $(V \neq \emptyset)$
2.        Construct a complete SLD-tree for $\leftarrow \mathcal{A}$ with respect to DDB.
3.        For every successful branch $i$:construct $\Delta_i = \{D \mid D \in EDB\}$
              and D is used as an input clause in branch $i$.
              Construct a branch i of an update tableau based on view
              for $IDB \cup EDB$ and delete request $\neg A$.
              Produce $IDB \cup EDB \backslash HS(i)$ as a result
4.        For every unsuccessful branch $j$:construct $\Delta_j = \{D \mid D \in EDB\}$
              and D is used as an input clause in branch $j$.
              Construct a branch j of an update magic set rule based on view
              for $IDB \cup EDB$ and insert request $A$.
              Produce $IDB \cup EDB \backslash HS(j)$ as a result
          Let $V := \{c \in IC \mid IDB \cup IC$ inconsistent with $\mathcal{A}$ with respect to $c$ $\}$
      return
5.    Produce $DDB$ as the result.
end.
```

Example 20.

$$\begin{array}{lllll}
IDB: & p \leftarrow a & & EDB: f \leftarrow & IC: \leftarrow b \\
 & q \leftarrow a & & \quad\;\; g \leftarrow & \\
 & p \leftarrow b \wedge f & & & \\
 & q \leftarrow b \wedge f & & & \\
 & p \leftarrow g \wedge a & & &
\end{array}$$

Suppose we want to insert p. First, we need to check consistency with IC and afterwards, we have to find Δ_i and Δ_j via tree deduction.

From algorithm 6 or 7 (only different is checking IC condition), the above example execute following steps:

```
Algorithm 7    View update algorithm based on Materialized view
Input :   A stratifiable (definite) deductive database $DDB = IDB \cup EDB \cup IC$
          an literals $\mathcal{A}$
Output:   A new stratifiable (definite) database $IDB \cup EDB' \cup IC$
begin
  1.  Construct a complete SLD-tree for $\leftarrow \mathcal{A}$ with respect to DDB.
  2.  For every successful branch $i$:construct $\Delta_i = \{D \mid D \in EDB\}$
        and D is used as an input clause in branch $i$.
  3.  For every unsuccessful branch $j$:construct $\Delta_j = \{D \mid D \in EDB\}$
        and D is used as an input clause in branch $j$.
  4.  Let $V := \{c \in IC \mid IDB \cup IC$ inconsistent with $\mathcal{A}$ with respect to $c$ $\}$
      While $(V \neq \emptyset)$
        Construct a branch i of an update tableau satisfying based on view
          for $IDB \cup EDB$ and delete request $A$.
        Produce $IDB \cup EDB \backslash HS(i)$ as a result
        Construct a branch j of an update magic set rule based on view
          for $IDB \cup EDB$ and insert request $A$.
        Produce $IDB \cup EDB \backslash HS(j)$ as a result
        Let $V := \{c \in IC \mid IDB \cup IC$ inconsistent with $\mathcal{A}$ with respect to $c$ $\}$
      return
  5.  Produce $DDB$ as the result.
end.
```

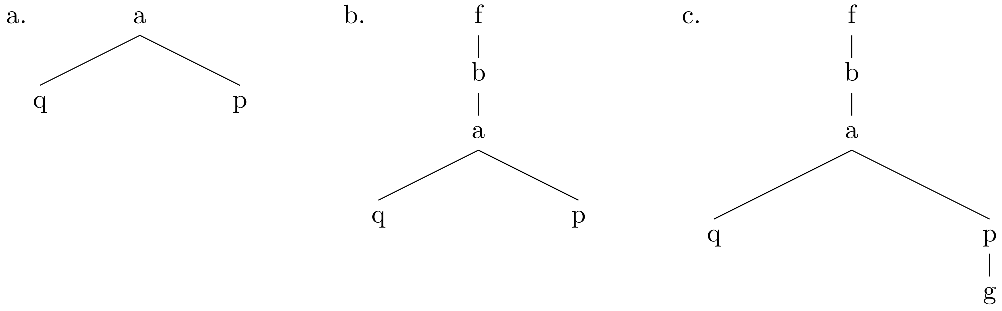
Step number with execution

(Input) $IDB : p \leftarrow a, q \leftarrow a, p \leftarrow b \wedge f, q \leftarrow b \wedge f, p \leftarrow g \wedge a$
$EDB : f \leftarrow, g \leftarrow$
$IC : \leftarrow b$

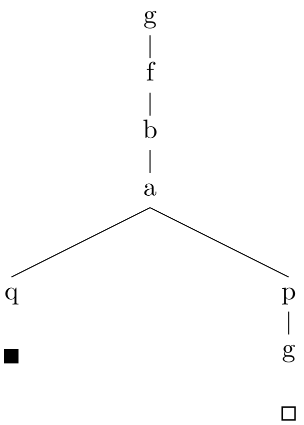
(0) $\{p \leftarrow a;\ q \leftarrow a;\ p \leftarrow b, f;\ q \leftarrow b, f;\ p \leftarrow g, a;\ b;\ g\}$

(1) $\{V = b\}$

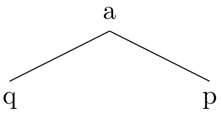
(2.1)



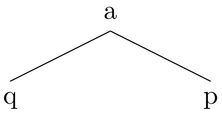
d.



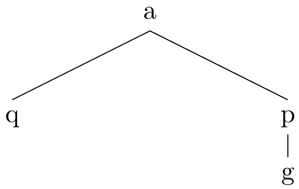
(2.2) a.



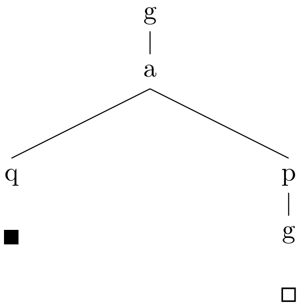
b.



c.



d.



(3-4) $\Delta_i = \{a, g\}$ and $\Delta_j = \{\}$
(5) $p \leftarrow a;\ q \leftarrow a;\ p \leftarrow g \wedge a;\ a,\ f,\ g$
(Output) $IDB : p \leftarrow a, q \leftarrow a, p \leftarrow g \wedge a$
$EDB' : a \leftarrow;\ f \leftarrow;\ g \leftarrow$
$IC :\leftarrow b$

This approach for view update may not satisfy (KB*7.1) in general. But, as shown in the sequel, conformation to (KB*7.3) is guaranteed and thus this approach results in rational update.

Lemma 5. *Let T be an update tableau based on a materialized view for $IDB \cup EDB$ and delete request $\neg A$ (A), Similarly, let M be an update magic set rule. Let S be the set of all EDB-closed locally minimal abductive explanations for A with respect to IDB. Let S' be the set of all EDB-cuts of T and EDB-seeds of M. Then, the following hold:*

- $S \subseteq S'$.
- $\forall \Delta' \in S' : \exists \Delta \in S$ *s.t.* $\Delta \subseteq \Delta'$.
- $\forall \Delta' \in S' : \Delta' \subseteq \bigcup S$.

Proof.

1. Consider a $\Delta(\Delta \in \Delta_i \cup \Delta_j) \in S$. We need to show that Δ is generated by algorithm 6 at step 2. From observation 4, it is clear that there exists a A-kernel X of DDB_G s.t. $X \cap EDB = \Delta_j$ and $X \cup EDB = \Delta_i$. Since $X \vdash A$, there must exist a successful derivation for A using only the elements of X as input clauses and similarly $X \nvdash A$. Consequently Δ must have been constructed at step 2.
2. Consider a $\Delta'((\Delta' \in \Delta_i \cup \Delta_j) \in S'$. Let Δ' be constructed from a successful(unsuccessful) branch i via $\Delta_i(\Delta_j)$. Let X be the set of all input clauses used in the refutation i. Clearly $X \vdash A(X \nvdash A)$. Further, there exists a minimal (with respect to set-inclusion) subset Y of X that derives A (i.e. no proper subset of Y derives A. Let $\Delta = Y \cap EDB$ $(Y \cup EDB)$. Since IDB does not (does) have any unit clauses, Y must contain some EDB facts, and so Δ is not empty (empty) and obviously $\Delta \subseteq \Delta'$. But, Y need not (need) be a A-kernel for IDB_G since Y is not ground in general. But it stands for several A-kernels with the same (different) EDB facts Δ in them. Thus, from observation 4, Δ is a DDB-closed locally minimal abductive explanation for A with respect to IDB_G and is contained in Δ'. minimal.

Lemma 6. *Let S and S' be sets of sets s.t.* $S \subseteq S'$ *and for every member X of* $S' \backslash S$*: X is a superset of some member of S and X is a subset of* $\bigcup S$*. Then, a set H is a hitting set for S iff it is a hitting set for S'*

Proof.

1. (**If part**) Given that H is a hitting set for S', we have to show that it is a hitting set for S too. First of all, observe that $\bigcup S = \bigcup S'$, and so $H \subseteq \bigcup S$. Moreover, by definition, for every non-empty member X of S', $H \cap X$ is not empty. Since $S \subseteq S'$, it follows that H is a hitting set for S too.

   (**Only if part**) Suppose H is a hitting set for S. As observed above, $H \subseteq \bigcup S'$. By definition, for every non-empty member $X \in S$, $X \cap H$ is not empty. Since every member of S' is a superset of some member of S, it is clear that H hits every member of S', and hence a hitting set for S' .

Lemma 7. *Let T and M as in Lemma 5. Then* $HS(b)$ *is a rational hitting set for A, for every open finished branch b in T (close finished branch b in M).*

Proof. Follows from the observation 4 (materialized view) in and (Behrend & Manthey 2008)

Theorem 16. *Algorithms 6 and 7 are rational, in the sense that they satisfy all the rationality postulates (KB*1) to (KB*6) and (KB*7.3).*

Proof. Follows from Observation 4,7 and Theorem 7.

8.4 Incomplete to Complete Information

Many of the proposals in the literature on incomplete databases have focussed on the extension of the relational model by the introduction of null values. In this section, we show how view update provides completion of incomplete information. More detailed surveys of this area can be found in (Meyden 1998).

The earliest extension of the relational model to incomplete information was that of Codd (Codd 1979) who suggested that missing values should be represented in tables by placing a special *null value* symbol $'*'$ at any table location for which the value is unknown. Table 5.3, shows an example of a database using this convention. Codd proposed an extension to the relational algebra for tables containing such nulls, based on three valued logic and a null substitution principle.

In terms of our general semantic scheme, the intended semantics of a database D consisting of Codd tables can be described by defining $Mod(D)$ to be the set of structures $M_{D'}$, where D' ranges over the relational databases obtained by replacing each occurrence of $'*'$ in the database D by some domain value. Different values may be substituted for different occurrences.

A plausible integrity constraint on the meaning of a relational operator on tables in $\mathcal{T}$ is that the result should be a table that represents the set of relations obtained by pointwise application of the operator on the models of these tables. For example, if R and S are tables in $\mathcal{T}$ then the result of the join $R \bowtie S$ should be equal to a table T in $\mathcal{T}$ such that

$$Mod(T) = \{r \bowtie t \mid r \in Mod(R),\ s \in Mod(S)\}$$

In case the definitions of the operators satisfy this integrity constraint (with respect to the definition of the semantics *Mod* on $\mathcal{T}$).

Let us consider what above equation requires if we take R and S to be the Codd Tables 5.3. First of all, note that in each model, if we take the value of the null in the tuple (delhibabu,*) to be v, then the join will contain one tuples (delhibabu, v), which include the value v. If T is to be a Codd table, it will need to contain tuples (delhibabu,X) to generate each of these tuples, where X are either constants or '*'. We now face a problem. First, X cannot be a constant c, for whatever the choice of c we can find an instance $r \in Mod(R)$ and $s \in Mod(S)$ for which the tuple (delhibabu, c) does not occur in $r \bowtie s$. If they were, X would have their values in models of T assigned independently.

Here the repetition of $*$ indicates that the *same* value is to be occurrence of the null in constructing a model of the table. Unfortunately, this extension does not suffice to satisfy the integrity constraint ($\forall x, y, z$ (y=x) $\leftarrow$ group_chair(x,y) $\wedge$ group_chair(x,z)).

Staff	Group
delhibabu	infor1
delhibabu	*

Group	Chair
infor1	mattias
*	aravindan

Table 3. Base Table after Transaction

In the model of these tables in which $* = infor1$, the join contains the tuple (delhibabu, infor1) and (infor1, aravindan).

$$\text{If } *_1 = \text{infor1 then (delhibabu, infor1)} \in R \bowtie S$$

$$\text{If } *_2 = \text{infor1 then (infor1, aravindan)} \in R \bowtie S$$

The following table shows when transaction is made to base table:

Staff	Group	Chair
delhibabu	infor1	mattias
delhibabu	*	aravindan

Table 4. $s \otimes r$ after Transaction

The following table shows completion of incomplete information with application of integrity constraint and redundancy:

Staff	Group	Chair
delhibabu	infor1	aravindan

Table 5. Redundant Table

8.5 A Comparative Study of view update algorithm and integrity constraint with our axiomatic method

During the process of updating database, two interrelated problems could arise. On one hand, when an update is applied to the database, integrity constraints could become inconsistent with request, then stop the process. On the other hand, when an update request consist on updating some derived predicate, a view update mechanism must be applied to translate the update request into correct updates on the underlying base facts. Our work focus on the integrity constraint maintenance approach. In this section, we extend Mayol and Teniente's (Mayol & Teniente 1999) survey for view update and integrity constraint.

The main aspects that must be taken into account during the process of view update and integrity constraint enforcement are the following: the problem addressed, the considered database schema, the allowed update requests, the used technique, update change and the obtained solutions. These six aspects provide the basic dimensions to be taken into account. We explain each dimension in this section and results are presented in Appendix.

Problem Addressed

(*Type*) - What kind of program to be used (stratified (S), Horn clause (H), Disjunctive database (D), Normal Logic program (N) and Other (O)).

(*View Update*) - Whether they are able to deal with view update or not (indicated by Yes or No in the second column in the appendix section).

(*integrity-constraint Enforcement*) - Whether they incorporate an integrity constraint checking (C)or an integrity constraint maintenance (M) or both apply (C-M) approach (indicated by check or maintain in the third column).

(*Run/Comp*) - Whether the method follows a run-time (transaction) or a compile-time approach (indicated by Run or Compile in the fourth column).

Database Schema Considered

(*Definition Language*) - The language mostly used is logic (L), although some methods use a relational language (R) and also uses an object-oriented (O-O).

(*The DB Schema Contains Views*) - All methods that deal with view update need views to be defined in the database schema. Some of other method allow to define views.

(*Restrictions Imposed on the Integrity Constraints*) - Some proposals impose certain restrictions on the kind of integrity constraints that can be defined and, thus, handled by their methods.

(*Static vs Dynamic Integrity Constraints*) - Integrity constraints may be either static (S), and impose restrictions involving only a certain state of the database, or dynamic (D).

Update Request Allowed

(*Multiple Update Request*) - An update request is multiple if it contains several updates to be applied together to the database.

(*Update Operators*) - Traditionally, three different basic update operators are distinguished: insertion (ι), deletion (δ) and modification (χ). Modification can always be simulated by a deletion followed by an insertion.

Update Processing Mechanism

(*Applied Technique*) - The techniques applied by these methods can be classified according to four different kinds of procedures, unfolding, SLD, active and predefined programs, respectively.

(*Taking Base Facts into Account* - Base facts can either be taken into account or not during update processing.

(*User Participation*) - User participation during update processing or not.

Update Changing Mechanism

(*Type of modification*) - Changing table by singleton like atom (S), sets of each types of modification(SS) and group of changes (G).
(*Changing Base Fact*) - Base fact can be changed either using principle of minimal change or complete change (maximal change).
(*Changing View Definition*) - Whether update process view definition is changed or not.

Obtained Solution

(*Our Axiom follow*) - When update process done, we are comparing our axiomatized method and which relevance policy holds ((KB*1) to (KB*6),(KB*7.1),(KB*7.2) and (KB*7.3) is enumerated 1 to 9)
(*Soundness*) - A method is correct if it only obtains solutions that satisfy the requested update, note NP mean Not Proved.
(*Completeness)* - A method is complete if it is able to obtain all solutions that satisfy a given update request.

Results of each method according to these features are summarized in Appendix.

9 Belief Update Vs Database Update

In this section we give overview of how belief update is related to database update. This section is motivated by works of Hansson's (Hansson 1991) and Keller's (Keller 1985)

9.1 View update vs Database update

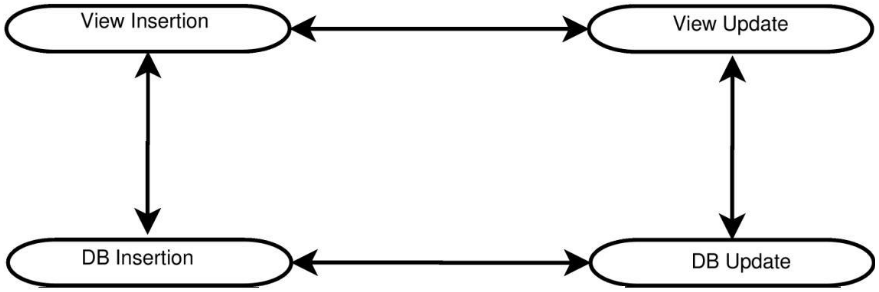


Fig. 5. View Update Vs Database Update

The view update problem exists already three decades Chen & Liao 2010 and Minker 1996. We are taking proof from Keller 1985, given a view definition of the question of choosing a view update translator arises.

This requires understanding the ways in which individual view update requests may be satisfied by database updates. Any particular view update request may result in a view state that does not correspond to any database state. Such a view update request may not be translated without relaxing the constraint which precludes view side effects. Otherwise, the update request is rejected by the view update translator. If we are lucky, there will be precisely one way to perform the database update that results in the desired view update. Since the view is many-to-one, the new view state may correspond to many database states. Of these database states, we would like to choose one that is "as close as possible", under some measure, to the original database state. That is, we would like to minimize the effect of the view update on the database.

9.2 Belief update vs Database update

If we look closely to the section (6.3 and 8.1), we easily find the following results. With evidence of Hansson's (Hansson 1991) and Liberatore (Liberatore & Schaerf 2004). Here BR and BU mean Belief Revision and Belief Update, respectively.

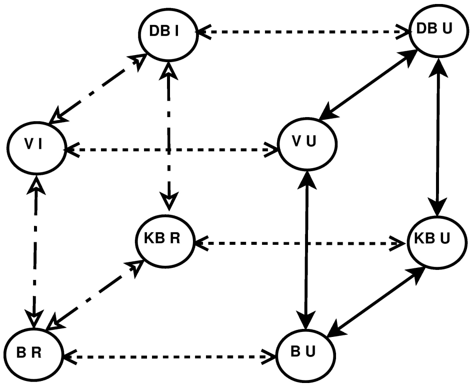


Fig. 6. Belief Update Vs Database Update

10 Abductive framework for Horn knowledge base dynamics

As discussed in Section 5, we introduced Horn knowledge base dynamics to deal with two important points: first, to handle belief states that need not be deductively closed; and the second point is the ability to declare certain parts of the belief as immutable. There is yet another, radically new approach to handle this problem, and this Section addresses this. In fact, this approach is very close to the Hansson's (Hansson 1992) dyadic representation of belief. In the similar abduction model by Boutilier & Beche 1995 and Pagnucco 1996 Here, we consider the immutable part as defining a new logical system. By a logical system, we mean that it defines its own consequence relation and closure operator. Based on this, we provide an abductive framework for Horn knowledge base dynamics.

A first order language consists of an alphabet $\mathcal{A}$ of a language $\mathcal{L}$. We assume a countable universe of variables Var, ranged over x,y,z, and a countable universe of relation (i.e. predicate) symbols, ranged over by $\mathcal{A}$ are finite. The following defines FOL, the language of first order logic with equality and binary relations:

$$\varphi ::= x = x \mid a(x, x) \mid \neg\varphi \mid \bigvee \phi \mid \bigwedge \phi \mid \exists X : \phi.$$

Here $\phi \subseteq FOL$ and $X \subseteq Var$ are finite sets of formulae and variables, respectively.

Definition 46 (Normal Logic Program (NLP) [22]). *By an alphabet $\mathcal{A}$ of a language $\mathcal{L}$ we mean disjoint sets of constants, predicate symbols, and function symbols, with at least one constant. In addition, any alphabet is assumed to contain a countably infinite set of distinguished variable symbols. A term over $\mathcal{A}$ is defined recursively as either a variable, a constant or an expression of the form $f(t_1, ..., t_n)$ where f is a function symbol of $\mathcal{A}$, n its arity, and the t_i are terms. An atom over $\mathcal{A}$ is an expression of the form $P(t_1, ..., t_n)$ where P is a predicate symbol of $\mathcal{A}$ and the t_i are terms. A literal is either an atom A or its default negation not A. We dub default literals those of the form not A. A term (atom or literal) is said ground if it does not contain variables. The set of all ground terms (atoms) of $\mathcal{A}$ is called the Herbrand universe (base) of A. A Normal Logic Program is a possibly infinite set of rules (with no infinite descending chains of syntactical dependency) of the form:*

$$H \leftarrow B_1, ..., B_n, not\ C_1, ..., not\ C_m,\ (with\ m, n \geq 0)$$

Where H,B_i and C_j are atoms, and each rule stands for all its ground instances. In conformity with the standard convention, we write rules of the form $H \leftarrow$ also simply as H (known as fact). An NLP P is called definite if none of its rules contain default literals. H is the head of the rule r, denoted by head(r), and body(r) denotes the set $\{B_1, ..., B_n, not\ C_1, ..., not\ C_m\}$ of all the literals in the body of r.

When doing problem modeling with logic programs, rules of the form

$$\perp \leftarrow B_1, ..., B_n, not\ C_1, ..., not\ C_m,\ (with\ m, n \geq 0)$$

with a non-empty body are known as a type of integrity constraints (ICs), specifically denials, and they are normally used to prune out unwanted candidate solutions. We abuse the *not* default negation notation applying it to non-empty sets of literals too: we write not S to denote $\{not\ s : s \in S\}$, and duality of $not\ not\ a \equiv a$. When S is an arbitrary, non-empty set of literals $S = \{B_1, ..., B_n, not\ C_1, ..., not\ C_m\}$ we use:

- S^+ denotes the set $\{B_1, \ldots, B_n\}$ of positive literals in S .
- S^- denotes the set $\{not\ C_1, \ldots, not\ C_m\}$ of negative literals in S .
- $|S| = S^+ \cup (not\ S^-)$ denotes the set $\{B_1, \ldots, B_n, C_1, \ldots, C_m\}$ of atoms of S.

As expected, we say a set of literals S is consistent iff $S^+ \cap |S^-| = \emptyset$. We also write $heads(P)$ to denote the set of heads of non-IC rules of a (possibly constrained) program P, i.e. $heads(P) = \{head(r) : r \in P\} \backslash \{\bot\}$, and $facts(P)$ to denote the set of facts of P - $facts(P) = \{head(r) : r \in P \wedge body(r) = \emptyset\}$.

Definition 47 (Level mapping[4]). *Let P be a normal logic program and B_P its Herbrand base. A* level *mapping for P is a function* $\|$: $B_P \rightarrow \mathbb{N}$ *of ground atoms to natural numbers. The mapping* $\|$ *is also extended to ground literals by assigning* $\mid \neg A \mid = \mid A \mid$ *for all ground atoms* $A \in B_P$. *For every ground literal L,* $\mid L \mid$ *is called as the* level *of L in P.*

Definition 48 (Acyclic program [4]). *Let P be a normal logic program and* $\|$ *a level mapping for P. P is called as acyclic with respect to* $\|$ *if for every ground clause* $H \leftarrow L_1, ..., L_n$ *(with* $n \geq 0$ *and* $finit$*) in P the level of A is higher then the level of every* L_i *(1* $\leq$*i*$\leq$ *n). Moreover P is called acyclic if P is acyclic with respect to some level mapping for P.*

Unlike Horn knowledge base dynamics, where knowledge is defined as a set of sentences, here we wish to define a Horn knowledge base KB with respect to a language $\mathcal{L}$, as an abductive framework $< P, Ab, IC, K >$, where,

* P is an acyclic normal logic program with all abducibles in P at level 0 and no non-abducible at level 0. P is referred to as a *logical system*. This in conjunction with the integrity constraints corresponds to immutable part of the Horn knowledge base, here P is defined by immutable part. This is discussed further in the next subsection;
* *Ab* is a set of atoms from $\mathcal{L}$, called the *abducibles*. This notion is required in an abductive framework, and this corresponds to the atoms that may appear in the updatable part of the knowledge;
* IC is the set of *integrity constraints,* a set of sentences from language $\mathcal{L}$. This specifies the integrity of a Horn knowledge base and forms a part of the knowledge that can not be modified over time;
* K is a set of sentences from $\mathcal{L}$. It is the *current knowledge*, and the only part of KB that changes over time. This corresponds to the updatable part of the Horn knowledge base. The main requirement here is that no sentence in K can have an atom that does not appear in *Ab*.

10.1 Logical system

The main idea of our approach is to consider the immutable part of the knowledge to define a new logical system. By a logical system, we mean that P defines its own consequence relation $\models_P$ and its closure Cn_p. Given P, we have the Herbrand Base HB_P and G_P, the ground instantiation of P.

An *abductive interpretation* I is a set of abducibles, i.e. $I \subseteq Ab$. How I interprets all the ground atoms of L [4] is defined, inductively on the level of atoms with respect to P, as follows:

* An atom A at level 0 (note that only abducibles are at level 0) is interpreted as: A is *true* in I iff $A \in I$, else it is *false* in I.
* An atom (literal) A at level n is interpreted as: A is true in I iff $\exists$ clause $A \leftarrow L_1, \ldots, L_n$ in G_P s.t. $\forall L_j$ $(1 \leq j \leq n)$ if L_j is an atom then L_j is true in I, else if L_j is a negative literal $\neg B_j$, then B_j is false in I.

This interpretation of ground atoms can be extended, in the usual way, to interpret sentences in L, as follows (where α and β are sentences):

* $\neg\alpha$ is true in I iff α is false in I.
* $\alpha \wedge \beta$ is true in I iff both α and β are true in I.
* $\alpha \vee \beta$ is true in I iff either α is true in I or β is true in I.
* $\forall\alpha$ is true in I iff all ground instantiations of α are true in I.
* $\exists\alpha$ is true in I iff some ground instantiation of α is true in I.

Given a sentence α in L, an abductive interpretation I is said to be an *abductive model* of α iff α is true in I. Extending this to a set of sentences K, I is a abductive model of K iff I is an abductive model of every sentence α in K.

Given a set of sentences K and a sentence α, α is said to be a *P-consequence* of K, written as $K \models_P \alpha$, iff every abductive model of K is an abductive model of α also. Putting it in other words, let $Mod(K)$ be the set of all abductive models of K. Then α is a P-consequence of K iff α is true in all abductive interpretations in $Mod(K)$. The *consequence operator* Cn_P is then defined as $Cn_P(K) = \{\alpha \mid K \models_P \alpha\} = \{\alpha \mid \alpha \text{ is true in all abductive interpretations in } Mod(K)\}$. K is said to be P-*consistent* iff there is no expression α s.t. $\alpha \in Cn_P(K)$ and $\neg\alpha \in Cn_P(K)$. Two sentences α and β are said to be *P-equivalent* to each other, written as $\alpha \equiv \beta$, iff they have the same set of abductive models , i.e. $Mod(\alpha) = Mod(\beta)$.

Properties of consequences operator

Since a new consequence operator is defined, it is reasonable, to ask whether it satisfies certain properties that are required in the Horn knowledge base dynamics context. Here, we observe that all the required properties, listed by various researchers in Horn knowledge base dynamics, are satisfied by the defined consequence operator. The following propositions follow from the above definitions, and can be verified easily.

[4] the set of all the ground atoms of L, in fact depends of L, and is given as HB_P, the Herbrand Base of P

Cn_P satisfies *inclusion, i.e.* $K \subseteq Cn_P(K)$.

Cn_P satisfies *iteration, i.e.* $Cn_P(K) = Cn_P(Cn_P(K))$.

Anther interesting property is *monotony,* i.e. if $K \subseteq K'$, then $Cn_P(K) \subseteq Cn_P(K')$. Cn_P satisfies monotony. To see this, first observe that $Mod(K') \subseteq Mod(K)$.

Cn_P satisfies *superclassicality* , i.e. if α can be derived from K by first order classical logic, then $\alpha \in Cn_P(K)$.

Cn_P satisfies *deduction* , i.e. if $\beta \in Cn_P(K \cup \{\alpha\})$, then $(\beta \leftarrow \alpha) \in Cn(K)$.

Cn_P satisfies *compactness* , i.e. if $\alpha \in Cn_P(K)$, then $\alpha \in Cn_P(K')$ for some finite subset K' of K.

Statics of a Horn knowledge base

The statics of a Horn knowledge base KB, is given by the current knowledge K and the integrity constraints IC. An abductive interpretation M is an abductive model of KB iff it is an abductive model of $K \cup IC$. Let $Mod(KB)$ be the set of all abductive models of KB. The *belief set* represented by KB, written as $KB^\bullet$ is given as,

$$KB^\bullet = Cn_P(K \cup IC) = \{\alpha | \alpha \text{ is true in every abductive model of } KB\}.$$

A belief (represented by a sentence in $\mathcal{L}$) α is *accepted* in KB iff $\alpha \in KB^\bullet$ (i.e. α is true in every model of KB). α is *rejected* in KB iff $\neg\alpha \in KB^\bullet$ (i.e. α is false in every model of KB). Note that there may exist a sentence α s.t. α is neither accepted nor rejected in KB (i.e. α is true in some but not all models of KB), and so KB represents a partial description of the world.

Two Horn knowledge bases KB_1 and KB_2 are said to be *equivalent* to each other, written as $KB_1 \equiv KB_2$, iff they are based on the same logical system and their current knowledge are P-equivalent, i.e. $P_1 = P_2,\ Ab_1 = Ab_2,\ IC_1 = IC_2$ and $K_1 \equiv K_2$. Obviously, two equivalent Horn knowledge bases KB_1 and KB_2 represent the same belief set, i.e. $KB_1^\bullet = KB_2^\bullet$.

10.2 Horn knowledge base dynamics

In AGM (Alchourron et al. 1985b) three kinds of belief dynamics are defined: expansion, contraction and revision. We consider all of them, one by one, in the sequel.

Expansion

Let α be new information that has to be added to a knowledge base KB. Suppose $\neg\alpha$ is not accepted in KB. Then, obviously α is P - consistent with IC, and KB can be *expanded* by α, by modifying K as follows:

$$KB + \alpha \equiv < P, Ab, IC, K \cup \{\alpha\} >$$

Note that we do not force the presence of α in the new K, but only say that α must be in the belief set represented by the expanded Horn knowledge base. If in case $\neg\alpha$ is accepted in KB (in other words, α is inconsistent with IC), then expansion of KB by α results in a inconsistent Horn knowledge base with no abductive models, i.e. $(KB + \alpha)^{\bullet}$ is the set of all sentences in $\mathcal{L}$.

Putting it in model-theoretic terms, KB can be expanded by a sentence α, when α is not false in all models of KB. The expansion is defined as:

$$Mod(KB + \alpha) = Mod(KB) \cap Mod(\alpha).$$

If α is false in all models of KB, then clearly $Mod(KB{+}\alpha)$ is empty, implying that expanded Horn knowledge base is inconsistent.

Revision

As usual, for revising and contracting a Horn knowledge base, the rationality of the change is discussed first. Later a construction is provided that complies with the proposed rationality postulates.

Rationality postulates

Let $KB =< P, Ab, IC, K >$ be revised by a sentence α to result in a new Horn knowledge base $KB \dotplus \alpha =< P', Ab', IC', K' >$.

When a Horn knowledge base is revised, we do not (generally) wish to modify the underlying logical system P or the set of abducibles *Ab*. This is refereed to as *inferential constancy* by Hansson (Hansson 1991 & 1992).

($\dotplus$1) (*Inferential constancy)* $P' = P$ *and* $Ab' = Ab$,$IC' = IC$.

($\dotplus$2) *(Success)*α is accepted in $KB \dotplus \alpha$, i.e. α is true in all models of $KB \dotplus \alpha$.

($\dotplus$3) (*Consistency)* α is satisfiable and P-consistent with IC iff $KB \dotplus \alpha$ is P-consistent, i.e. $Mod(\{\alpha\} \cup IC)$ is not empty iff $Mod(KB \dotplus \alpha)$ is not empty.

($\dotplus$4) (*Vacuity)* If $\neg\alpha$ is not accepted in KB, then $KB \dotplus \alpha \equiv KB + \alpha$, i.e. if α is not false in all models of KB, then $Mod(KB \dotplus \alpha) = Mod(KB) \cap Mod(\alpha)$.

($\dotplus$5) (*Preservation)*If $KB \equiv KB'$ and $\alpha \equiv \beta$, then $KB \dotplus \alpha \equiv KB' \dotplus \beta$, i.e. if $Mod(KB) = Mod(KB')$ and $Mod(\alpha) = Mod(\beta)$, then $Mod(KB \dotplus \alpha) = Mod(KB \dotplus \beta)$.

($\dotplus$6) (*Extended Vacuity 1)*$(KB \dotplus \alpha) + \beta$ implies $KB \dotplus (\alpha \wedge \beta)$, i.e. $(Mod(KB \dotplus \alpha) \cap Mod(\beta)) \subseteq Mod(KB \dotplus (\alpha \wedge \beta))$.

($\dotplus$7) (*Extended Vacuity 2)*If $\neg\beta$ is not accepted in $(KB \dotplus \alpha)$, then $KB \dotplus (\alpha \wedge \beta)$ implies $(KB \dotplus \alpha) + \beta$, i.e. if β is not false in all models of $KB \dotplus \alpha$, then $Mod(KB \dotplus (\alpha \wedge \beta)) \subseteq (Mod(KB \dotplus \alpha) \cap Mod(\beta))$.

Construction

Let $\mathcal{S}$ stand for the set of all abductive interpretations that are consistent with IC, i.e. $\mathcal{S} = Mod(IC)$. We do not consider abductive interpretations that are not models of IC, simply because IC does not change during revision. Observe that when IC is empty, $\mathcal{S}$ is the set of all abductive interpretations. Given a Horn knowledge base KB, and two abductive interpretations I_1 and I_2 from $\mathcal{S}$, we can compare how close these interpretations are to KB by using an order $\leq_{KB}$ among abductive interpretations in $\mathcal{S}$. $I_1 <_{KB} I_2$ iff $I_1 \leq_{KB} I_2$ and $I_2 \nleq_{KB} I_1$.

Let $\mathcal{F} \subseteq \mathcal{S}$. An abductive interpretation $I \in \mathcal{F}$ is minimal in $\mathcal{F}$ with respect to $\leq_{KB}$ if there is no $I' \in \mathcal{F}$ s.t. $I' <_{KB} I$. Let, $Min(\mathcal{F}, \leq_{KB}) =$ $\{I \mid I$ is minimal in
$\mathcal{F}$ with respect to $\leq_{KB}\}$.

For any Horn knowledge base KB, the following are desired properties of $\leq_{KB}$:

($\leq$ 1) (*Pre-order)*$\leq_{KB}$ is a *pre-order* , i.e. it is transitive and reflexive.
($\leq$ 2) (*Connectivity)*$\leq_{KB}$ is *total* in $\mathcal{S}$, i.e. $\forall I_1, I_2 \in \mathcal{S}$: either $I_1 \leq_{KB} I_2$ or $I_2 \leq_{KB} I_1$.
($\leq$ 3) (*Faithfulness)*$\leq_{KB}$ is *faithful* to KB, i.e. $I \in Min(\mathcal{S}, \leq_{KB})$ iff $I \in Mod(KB)$.
($\leq$ 4) (*Minimality)*For any non-empty subset $\mathcal{F}$ of $\mathcal{S}$, $Min(\mathcal{F}, \leq_{KB})$ is not empty.
($\leq$ 5) (*Preservance)]* For any Horn knowledge base KB', if $KB \equiv KB'$ then $\leq_{KB} = \leq_{KB'}$.

Let KB (and consequently K) be revised by a sentence α, and $\leq_{KB}$ be a rational order that satisfies ($\leq$ 1) to ($\leq$ 5). Then the abductive models of the revised Horn knowledge base are given precisely by: $Min(Mod(\{\alpha\} \cup IC), \leq_{KB})$. Note that, this construction does not say what the resulting K is, but merely says what should be the abductive models of the new Horn knowledge base.

Representation theorem

Now, we proceed to show that revision of KB by α, as constructed above, satisfies all the rationality postulates stipulated in the beginning of this section. This is formalized by the following lemma.

Lemma 8. *Let KB be a Horn knowledge base, $\leq_{KB}$ an order among $\mathcal{S}$ that satisfies* ($\leq$ 1) *to* ($\leq$ 5)*. Let a revision operator $\dotplus$ be defined as: for any sentence α, $Mod(KB \dotplus \alpha) = Min(Mod(\{\alpha\} \cup IC), \leq_{KB})$. Then $\dotplus$ satisfies all the rationality postulates for revision* ($\dotplus$1) *to* ($\dotplus$7)*.*

Proof($\dotplus$1) $P' = P$ and $Ab' = Ab$ and $IC' = IC$
This is satisfied obviously, since our construction does not touch P and Ab, and IC follows from every abductive interpretation in $Mod(KB \dotplus \alpha)$.

($\dotplus$2) α is accepted in $KB \dotplus \alpha$
Note that every abductive interpretation $M \in Mod(KB + \alpha)$ is a model of α. Hence α is accepted in $KB \dotplus \alpha$.

($\dotplus$3) α is satisfiable and P-consistent with IC iff $KB \dotplus \alpha$ is P-consistent.
If part: If $KB \dotplus \alpha$ is P-consistent , then $Mod(KB \dotplus \alpha)$ is not empty. This implies that $Mod(\{\alpha\} \cup IC)$ is not empty, and hence α is satisfiable and P-consistent with IC.
Only if part: If α is satisfiable and P-consistent with IC, then $Mod(\{\alpha\} \cup IC)$ is not empty, and ($\leq$ 4) ensures that $Mod(KB \dotplus \alpha)$ is not empty. Thus, $KB \dotplus \alpha$ is P-consistent.

($\dotplus$4) If $\neg\alpha$ is not accepted in KB, then $KB \dotplus \alpha \equiv KB + \alpha$.
We have to establish that $Min(Mod(\{\alpha\} \cup IC), \leq_{KB}) = Mod(KB) \cap Mod(\alpha)$. Since $\neg\alpha$ is not accepted in KB, $Mod(KB) \cap Mod(\alpha)$ is not empty. The required result follows immediately from the fact that $\leq_{KB}$ is faithful to KB (i.e. satisfies ≤ 3), which selects only and all those models of α which are also models of KB.

($\dotplus$5) If $KB \equiv KB'$ and $\alpha \equiv \beta$ then $KB \dotplus \alpha = KB' \dotplus \beta$
($\leq$ 5) ensures that $\leq_{KB} = \leq_{KB'}$. The required result follows immediately from this and the fact that $Mod(\alpha) = Mod(\beta)$.

($\dotplus$6) $(KB \dotplus \alpha) + \beta$ implies $KB \dotplus (\alpha \wedge \beta)$.
We consider this in two cases. When $\neg\beta$ is accepted in $KB \dotplus \alpha$, $(KB \dotplus \alpha) + \beta$ is the set of all sentences from $\mathcal{L}$, and the postulate follows immediately. Instead when $\neg\beta$ is not accepted in $KB \dotplus \alpha$, this postulates coincides with the next one.

($\dotplus$7) If $\neg\beta$ is not accepted in $KB \dotplus \alpha$, then $KB \dotplus (\alpha \wedge \beta)$ implies $(KB \dotplus \alpha) + \beta$.
Together with the second case of previous postulate, we need to show that $KB \dotplus (\alpha \wedge \beta) = (KB \dotplus \alpha) + \beta$. In other words, we have to establish that $Min(Mod(\{\alpha \wedge \beta\} \cup IC), \leq_{KB}) = Mod(KB \dotplus \alpha) \cap Mod(\beta)$. For the sake of simplicity, let us represent $Min(Mod(\{\alpha \wedge \beta\} \cup IC), \leq_{KB})$ by P, and $Mod(KB \dotplus \alpha) \cap Mod(\beta)$, which is the same as $Min(Mod(\{\alpha\} \cup IC), \leq_{KB}) \cap Mod(\beta)$, by Q. The required result is obtained in two parts:

1) $\forall$ (abductive interpretation)M: if $M \in P$, then $M \in Q$
Obviously $M \in Mod(\beta)$. Assume that $M \notin Min(Mod(\{\alpha\} \cup IC), \leq_{KB})$. This can happen in two cases, and we show that both the cases lead to contradiction.
Case A: No model of β is selected by $\leq_{KB}$ from $Mod(\{\alpha\} \cup IC)$. But this contradicts our initial condition that $\neg\beta$ is not accepted in $KB \dotplus \alpha$.
Case B: Some model, say M', of β is selected by $\leq_{KB}$ from $Mod(\{\alpha\} \cup IC)$. Since M is not selected, it follows that $M' <_{KB} M$. But then this contradicts our initial assumption that $M \in P$. So, $P \subseteq Q$.
2) $\forall$ (abductive interpretation)M: if $M \in Q$, then $M \in P$
$M \in Q$ implies that M is a model of both α and β, and M is selected by $\leq_{KB}$ from $Mod(\{\alpha\} \cup IC)$. Note that $Mod(\{\alpha \wedge \beta\} \cup IC) \subseteq Mod(\{\alpha\} \cup IC)$. Since M is selected by $\leq_{KB}$ in a bigger set (i.e. $Mod(\{\alpha\} \cup IC)$), $\leq_{KB}$ must select M from its subset $Mod(\{\alpha \wedge \beta\} \cup IC)$ also. Hence $Q \subseteq P$.

But, that is not all. Any rational revision of KB by α, that satisfies all the rationality postulates, can be constructed by our construction method, and this is formalized below.

Lemma 9. *Let KB be a Horn knowledge base and $\dotplus$ a revision operator that satisfies all the rationality postulates for revision* ($\dotplus$1) *to* ($\dotplus$7)*. Then, there exists an order $\leq_{KB}$ among $\mathcal{S}$, that satisfies* ($\leq$ 1) *to* ($\leq$ 5)*, and for any sentence α, $Mod(KB \dotplus \alpha)$ is given in $Min(Mod(\{\alpha\} \cup IC), \leq_{KB})$.*

Proof. Let us construct an order $\leq_{KB}$ among interpretations in $\mathcal{S}$ as follows: For any two abductive interpretations I and I' in $\mathcal{S}$, define $I \leq_{KB} I'$ iff either

$I \in Mod(KB)$ or $I \in Mod(KB \dot{+} form(I, I'))$, where $form(I, I')$ stands for sentence whose only models are I and I'. We will show that $\leq_{KB}$ thus constructed satisfies (≤ 1) to (≤ 5) and $Min(Mod(\{\alpha\} \cup IC), \leq_{KB}) = Mod(KB \dot{+} \alpha)$.

First, we show that $Min(Mod(\{\alpha\} \cup IC), \leq_{KB}) = Mod(KB \dot{+} \alpha)$.Suppose α is not satisfiable, i.e. $Mod(\alpha)$ is empty, or α does not satisfy IC, then there are no abductive models of $\{\alpha\} \cup IC$, and hence $Min(Mod(\{\alpha\} \cup IC), \leq_{KB})$ is empty. From $(\dot{+}3)$, we infer that $Mod(KB \dot{+} \alpha)$ is also empty. When α is satisfiable and α satisfies IC, the required result is obtained in two parts:

1) *If $I \in Min(Mod(\{\alpha\} \cup IC), \leq_{KB})$, then $I \in Mod(KB \dot{+} \alpha)$*
   Since α is satisfiable and consistent with IC, $(\dot{+}3)$ implies that there exists at least one model, say I', for $KB \dot{+} \alpha$. From $(\dot{+}1)$, it is clear that I' is a model of IC, from $(\dot{+}2)$ we also get that I' is a model of α, and consequently $I \leq_{KB} I'$ (because $I \in Min(Mod(\{\alpha\} \cup IC), \leq_{KB})$). Suppose $I \in Mod(KB)$, then $(\dot{+}4)$ immediately gives $I \in Mod(KB \dot{+} \alpha)$. If not, from our definition of $\leq_{KB}$, it is clear that $I \in Mod(KB \dot{+} form(I, I'))$. Note that $\alpha \wedge form(I, I') \equiv form(I, I')$, since both I and I' are models of α. From $(\dot{+}6)$ and $(\dot{+}7)$, we get $Mod(KB \dot{+} \alpha) \cap \{I, I'\} = Mod(KB \dot{+} form(I, I'))$. Since $I \in Mod(KB \dot{+} form(I, I'))$, it immediately follows that $I \in Mod(KB \dot{+} \alpha)$.
2) *If $I \in Mod(KB \dot{+} \alpha)$, then $I \in Min(Mod(\{\alpha\} \cup IC), \leq_{KB})$.*
   From $(\dot{+}1)$ we get I is a model of IC, and from $(\dot{+}2)$, we obtain $I \in Mod(\alpha)$. Suppose $I \in Mod(KB)$, then from our definition of $\leq_{KB}$, we get $I \leq_{KB} I'$, for any other model I' of α and IC, and hence $I \in Min(Mod(\{\alpha\} \cup IC), \leq_{KB})$. Instead, if I is not a model of KB, then, to get the required result, we should show that $I \in Mod(KB \dot{+} form(I, I'))$, for every model I' of α and IC. As we have observed previously, from $(\dot{+}6)$ and $(\dot{+}7)$, we get $Mod(KB \dot{+} \alpha) \cap \{I, I'\} = Mod(KB \dot{+} form(I, I'))$. Since $I \in Mod(KB \dot{+} \alpha)$, it immediately follows that $I \in Mod(KB \dot{+} form(I, I'))$. Hence $I \leq_{KB} I'$ for any model I' of α and IC, and consequently, $I \in Min(Mod(\{\alpha\} \cup IC), \leq_{KB}$).

Now we proceed to show that the order $\leq_{KB}$ among $\mathcal{S}$, constructed as per our definition, satisfies all the rationality axioms (≤ 1) to (≤ 5).

(≤ 1) *$\leq_{KB}$ is a pre-order.*
Note that we need to consider only abductive interpretations from $\mathcal{S}$. From $(\dot{+}2)$ and $(\dot{+}3)$, we have $Mod(KB \dot{+} form(I, I')) = \{I\}$, and so $I \leq_{KB} I$. Thus $\leq_{KB}$ satisfies reflexivity. let $I_1 \in Mod(IC)$ and $I_2 \notin Mod(IC)$. Clearly, it is possible that two interpretations I_1 and I_2 are not models of KB, and $Mod(KB \dot{+} form(I_1, I_2)) = \{I_1\}$. So, $I_1 \leq_{KB} I_2$ does not necessarily imply $I_2 \leq_{KB} I_1$, and thus $\leq_{KB}$ satisfies anti-symmetry.
To show the transitivity, we have to prove that $I_1 \leq_{KB} I_3$, when $I_1 \leq_{KB} I_2$ and $I_2 \leq_{KB} I_3$ hold. Suppose $I_1 \in Mod(KB)$, then $I_1 \leq_{KB} I_3$ follows immediately from our definition of $\leq_{KB}$. On the other case, when $I_1 \notin Mod(KB)$, we first observe that $I_1 \in Mod(KB \dot{+} form(I_1, I_2))$, which follows from definition of $\leq_{KB}$ and $I_1 \leq_{KB} I_2$. Also observe that $I_2 \notin Mod(KB)$. If I_2 were

a model of KB, then it follows from $(\dotplus 4)$ that $Mod(KB \dotplus form(I_1, I_2)) = Mod(KB) \cap \{I_1, I_2\} = \{I_2\}$, which is a contradiction, and so $I_2 \notin Mod(KB)$. This, together with $I_2 \leq_{KB} I_3$, implies that $I_2 \in Mod(KB \dotplus form(I_2, I_3))$. Now consider $Mod(KB \dotplus form(I_1, I_2, I_3))$. Since $\dotplus$ satisfies $(\dotplus 2)$ and $(\dotplus 3)$, it follows that this is a non-empty subset of $\{I_1, I_2, I_3\}$. We claim that $Mod(KB \dotplus form(I_1, I_2, I_3)) \cap \{I_1, I_2\}$ can not be empty. If it is empty, then it means that $Mod(KB \dotplus form(I_1, I_2, I_3)) = \{I_3\}$. Since $\dotplus$ satisfies $(\dotplus 6)$ and $(\dotplus 7)$, this further implies that $Mod(KB \dotplus form(I_2, I_3)) = Mod(KB \dotplus form(I_1, I_2, I_3)) \cap \{I_2, I_3\} = \{I_3\}$. This contradicts our observation that $I_2 \in Mod(KB \dotplus form(I_2, I_3))$, and so $Mod(KB \dotplus form(I_1, I_2, I_3)) \cap \{I_1, I_2\}$ can not be empty. Using $(\dotplus 6)$ and $(\dotplus 7)$ again, we get $Mod(KB \dotplus form(I_1, I_2)) = Mod(KB \dotplus form(I_1, I_2, I_3)) \cap \{I_1, I_2\}$. Since we know that $I_1 \in Mod(KB \dotplus form(I_1, I_2))$, it follows that $I_1 \in Mod(KB \dotplus form(I_1, I_2, I_3))$. From $(\dotplus 6)$ and $(\dotplus 7)$ we also get $Mod(KB \dotplus form(I_1, I_3)) = Mod(KB \dotplus form(I_1, I_2, I_3)) \cap \{I_1, I_3\}$, which clearly implies that $I_1 \in Mod(KB \dotplus form(I_1, I_3))$. From our definition of $\leq_{KB}$, we now obtain $I_1 \leq_{KB} I_3$. Thus, $\leq_{KB}$ is a pre-order.

(≤ 2) *$\leq_{KB}$ is total.*
Since $\dotplus$ satisfies $(\dotplus 2)$ and $(\dotplus 3)$, for any two abductive interpretations I and I' in $\mathcal{S}$, it follows that $Mod(KB \dotplus form(I, I'))$ is a non-empty subset of $\{I, I'\}$. Hence, $\leq_{KB}$ is total.

(≤ 3) *$\leq_{KB}$ is faithful to KB.*
From our definition of $\leq_{KB}$, it follows that $\forall I_1, I_2 \in Mod(KB) : I_1 <_{KB} I_2$ does not hold. Suppose $I_1 \in Mod(KB)$ and $I_2 \notin Mod(KB)$. Then, we have $I_1 \leq_{KB} I_2$. Since $\dotplus$ satisfies $(\dotplus 4)$, we also have $Mod(KB \dotplus form(I_1, I_2)) = \{I_1\}$. Thus, from our definition of $\leq_{KB}$, we can not have $I_2 \leq_{KB} I_1$. So, if $I_1 \in Mod(KB)$ and $I_2 \notin Mod(KB)$, then $I_1 <_{KB} I_2$ holds. Thus, $\leq_{KB}$ is faithful to KB.

(≤ 4) *For any non-empty subset $\mathcal{F}$ of $\mathcal{S}$, $Min(\mathcal{F}, \leq_{KB})$ is not empty.*
Let α be a sentence such that $Mod(\{\alpha\} \cup IC) = \mathcal{F}$. We have already shown that $Mod(KB \dotplus \alpha) = Min(\mathcal{F}, \leq_{KB})$. Since, $\dotplus$ satisfies $(\dotplus 3)$, it follows that $Mod(KB \dotplus \alpha)$ is not empty, and thus $Min(\mathcal{F}, \leq_{KB})$ is not empty.

(≤ 5) *If $KB \equiv KB'$, then $\leq_{KB} = \leq_{KB'}$.*
This follows immediately from the fact that $\dotplus$ satisfies $(\dotplus 5)$.

Thus, the order among interpretations $\leq_{KB}$, constructed as per our definition, satisfies (≤ 1) to (≤ 5), and $Mod(KB \dotplus \alpha) = Min(Mod(\{\alpha\} \cup IC), \leq_{KB})$.

So, we have a one to one correspondence between the axiomatization and the construction, which is highly desirable, and this is summarized by the following *representation theorem.*

Theorem 17. *Let KB be revised by α, and $KB \dotplus \alpha$ be obtained by the construction discussed above. Then, $\dotplus$ is a revision operator iff it satisfies all the rationality postulates $(\dotplus 1)$ to $(\dotplus 7)$.*

Proof. Follows from Lemma 8 and Lemma 9

Contraction

Contraction of a sentence from a Horn knowledge base KB is studied in the same way as that of revision. We first discuss the rationality of change during contraction and proceed to provide a construction for contraction using duality between revision and contraction.

Rationality Postulates

Let $KB =< P, Ab, IC, K >$ be contracted by a sentence α to result in a new Horn knowledge base $KB\dot{-}\alpha =< P', Ab', IC', K' >$.

($\dot{-}$1) (*Inferential Constancy)*$P' = P$ and $Ab' = Ab$ and $IC' = IC$.

($\dot{-}$2) (*Success)*If $\alpha \notin Cn_P(KB)$, then α is not accepted in $KB\dot{-}\alpha$, i.e. if α is not true in all the abductive interpretations, then α is not true in all abductive interpretations in $Mod(KB\dot{-}\alpha)$.

($\dot{-}$3) (*Inclusion)*$\forall$ (belief) β:if β is accepted in $KB\dot{-}\alpha$, then β is accepted in KB, i.e. $Mod(KB) \subseteq Mod(KB\dot{-}\alpha)$.

($\dot{-}$4) (*Vacuity)*If α is not accepted in KB, then $KB\dot{-}\alpha = KB$, i.e. if α is not true in all the abductive models of KB, then $Mod(KB\dot{-}\alpha) = Mod(KB)$.

($\dot{-}$5) (*Recovery)*$(KB\dot{-}\alpha)+\alpha$ implies KB, i.e. $Mod(KB\dot{-}\alpha)\cap Mod(\alpha) \subseteq Mod(KB)$.

($\dot{-}$6) (*Preservation)*If $KB \equiv KB'$ and $\alpha \equiv \beta$, then $KB\dot{-}\alpha = KB'\dot{-}\beta$, i.e. if $Mod(KB) = Mod(KB')$ and $Mod(\alpha) = Mod(\beta)$, then $Mod(KB\dot{-}\alpha) = Mod(KB'\dot{-}\beta)$.

($\dot{-}$7) (*Conjunction 1)* $KB\dot{-}(\alpha \wedge \beta)$ implies $KB\dot{-}\alpha \cap KB\dot{-}\beta$, i.e. $Mod(KB\dot{-}(\alpha \wedge \beta)) \subseteq Mod(KB\dot{-}\alpha) \cup Mod(KB\dot{-}\beta)$.

($\dot{-}$8) (*Conjunction 2)*If α is not accepted in $KB\dot{-}(\alpha \wedge \beta)$, then $KB\dot{-}\alpha$ implies $KB\dot{-}(\alpha \wedge \beta)$, i.e. if α is not true in all the models of $KB\dot{-}(\alpha \wedge \beta)$, then $Mod(KB\dot{-}\alpha) \subseteq Mod(KB\dot{-}(\alpha \wedge \beta))$.

Before providing a construction for contraction, we wish to study the duality between revision and contraction. The Levi and Harper identities still holds in our case, and is discussed in the sequel.

Relationship between contraction and revision

Contraction and revision are related to each other. Given a contraction function $\dot{-}$, a revision function $\dot{+}$ can be obtained as follows:

$$(\textit{Levi Identity}) \qquad Mod(KB \dot{+} \alpha) = Mod(KB\dot{-}\neg\alpha) \cap Mod(\alpha)$$

The following theorem formally states that Levi identity holds in our approach.

Theorem 18. *Let $\dot{-}$ be a contraction operator that satisfies all the rationality postulates* ($\dot{-}$1) *to* ($\dot{-}$8)*. Then, the revision function $\dot{+}$, obtained from $\dot{-}$ using the Levi Identity, satisfies all the rationality postulates* ($\dot{+}$1) *to* ($\dot{+}$7)*.*

Proof. Let $KB =< P, Ab, IC, K >$ be contracted by a sentence α to result in a new Horn knowledge base $KB\dot{-}\alpha =< P', Ab', IC', K' >$.

($\dotplus$1) (*Inferential constancy)* $P' = P$ *and* $Ab' = Ab$,$IC' = IC$.

($\dotplus$2) *(Success)*α is accepted in $KB\dotplus\alpha$, i.e. α is true in all models of $(Mod(KB\dot{-}\neg\alpha)\cap Mod(\alpha))$.

($\dotplus$3) (*Consistency)* α is satisfiable and P-consistent with IC iff $KB \dotplus \alpha$ is P-consistent, i.e. $(Mod(\{\alpha\} \cup IC))$ is not empty iff $Mod(KB\dot{-}\neg\alpha) \cap Mod(\alpha)$ is not empty.

($\dotplus$4) (*Vacuity)* If $\neg\alpha$ is not accepted in KB, then $KB \dotplus \alpha \equiv KB + \alpha$, i.e. if α is not false in all models of KB, then $(Mod(KB \dotplus \alpha)) = (Mod(KB\dot{-}\neg\alpha) \cap Mod(\alpha)) \cap Mod(\alpha)$

($\dotplus$5) (*Preservation)*If $KB \equiv KB'$ and $\alpha \equiv \beta$, then $KB \dotplus \alpha \equiv KB' \dotplus \beta$, i.e. if $Mod(KB) = Mod(KB')$ and $Mod(\alpha) = Mod(\beta)$, then $(Mod(KB\dot{-}\neg\alpha) \cap Mod(\alpha)) = (Mod(KB\dot{-}\neg\beta) \cap Mod(\beta))$.

($\dotplus$6) (*Extended Vacuity 1)*$(KB \dotplus \alpha) + \beta$ implies $((KB\dot{-}\neg\alpha) \cap (\alpha)) \wedge \beta)$, i.e. $(Mod(KB \dotplus \alpha) \cap Mod(\beta)) \subseteq Mod(KB \dotplus (\alpha \wedge \beta))$.

($\dotplus$7) (*Extended Vacuity 2)*If $\neg\beta$ is not accepted in $(KB \dotplus \alpha)$, then $((KB\dot{-}\neg\alpha) \cap (\alpha)) \wedge \beta)$ implies $(KB \dotplus \alpha) + \beta$, i.e. if β is not false in all models of $KB \dotplus \alpha$, then $Mod(KB \dotplus (\alpha \wedge \beta)) \subseteq (Mod(KB \dotplus \alpha) \cap Mod(\beta))$.

Similarly, a contraction function $\dot{-}$ can be constructed using the given revision function $\dotplus$ as follows:

$$(\textit{Harper Identity}) \qquad Mod(KB\dot{-}\alpha) = Mod(KB) \cup Mod(KB \dotplus \neg\alpha)$$

Theorem 19. *Let $\dotplus$ be a revision operator that satisfies all the rationality postulates* ($\dotplus$1) *to* ($\dotplus$7)*. Then, the contraction function $\dot{-}$, obtained from $\dotplus$ using the Harper Identity, satisfies all the rationality postulates* ($\dot{-}$1) *to* ($\dot{-}$8)*.*

Proof. Let $KB =< P, Ab, IC, K >$ be contracted by a sentence α to result in a new Horn knowledge base $KB\dot{-}\alpha =< P', Ab', IC', K' >$.

($\dot{-}$1) (*Inferential Constancy)*$P' = P$ and $Ab' = Ab$ and $IC' = IC$.

($\dot{-}$2) (*Success)*If $\alpha \notin Cn_P(KB)$, then α is not accepted in $KB\dot{-}\alpha$, i.e. if α is not true in all the abductive interpretations, then α is not true in all abductive interpretations in $Mod(KB) \cup Mod(KB \dotplus \neg\alpha)$.

($\dot{-}$3) (*Inclusion)*$\forall$ (belief) β:if β is accepted in $KB\dot{-}\alpha$, then β is accepted in KB, i.e. $Mod(KB) \subseteq (Mod(KB) \cup Mod(KB \dotplus \neg\alpha))$.

($\dot{-}$4) (*Vacuity)*If α is not accepted in KB, then $KB\dot{-}\alpha = KB$, i.e. if α is not true in all the abductive models of KB, then $Mod(KB\dot{-}\alpha) = Mod(KB)$.

($\dot{-}$5) (*Recovery)*$(KB\dot{-}\alpha)+\alpha$ implies KB, i.e. $(Mod(KB\dot{-}\alpha)\cap Mod(\alpha)) \subseteq Mod(KB)$.

($\dot{-}$6) (*Preservation)*If $KB \equiv KB'$ and $\alpha \equiv \beta$, then $KB\dot{-}\alpha = KB'\dot{-}\beta$, i.e. if $Mod(KB) = Mod(KB')$ and $Mod(\alpha) = Mod(\beta)$, then $(Mod(KB) \cup Mod(KB \dotplus \neg\alpha)) = (Mod(KB) \cup Mod(KB \dotplus \neg\beta))$.

($\dot{-}$7) (*Conjunction 1)* $KB\dot{-}(\alpha \wedge \beta)$ implies $KB\dot{-}\alpha \cap KB\dot{-}\beta$, i.e. $Mod(KB\dot{-}(\alpha \wedge \beta)) \subseteq (Mod(KB) \cup Mod(KB \dotplus \neg\alpha)) \cup (Mod(KB) \cup Mod(KB \dotplus \neg\beta))$.

($\dot{-}$8) (*Conjunction 2)*If α is not accepted in $KB\dot{-}(\alpha \wedge \beta)$, then $KB\dot{-}\alpha$ implies $KB\dot{-}(\alpha \wedge \beta)$, i.e. if α is not true in all the models of $KB\dot{-}(\alpha \wedge \beta)$, then $(Mod(KB) \cup Mod(KB \dotplus \neg\alpha)) \subseteq Mod(KB\dot{-}(\alpha \wedge \beta))$.

Construction

Given the construction for revision, based on order among interpretation in $\mathcal{S}$, a construction for contraction can be provided as:

$$Mod(KB\dot{-}\alpha) = Mod(KB) \cup Min(Mod(\{\neg\alpha\} \cup IC), \leq_{KB}),$$

where $\leq_{KB}$ is the relation among interpretations in $\mathcal{S}$ that satisfies the rationality axioms (≤ 1) to (≤ 5). As in the case of revision, this construction says what should be the models of the resulting Horn knowledge base, and does not explicitly say what the resulting Horn knowledge base is.

Representation theorem

Since the construction for contraction is based on a rational contraction for revision, the following lemma and theorem follow obviously.

Lemma 10. *Let KB be a Horn knowledge base, $\leq_{KB}$ an order among $\mathcal{S}$ that satisfies (≤ 1) to (≤ 5). Let a contraction operator $\dot{-}$ be defined as: for any sentence α, $Mod(KB\dot{-}\alpha) = Mod(KB) \cup Min(Mod(\{\neg\alpha\} \cup IC), \leq_{KB})$. Then $\dot{-}$ satisfies all the rationality postulates for contraction $(\dot{-}1)$ to $(\dot{-}8)$.*

Proof. *Follows from Theorem 17 and Theorem 19*

Lemma 11. *Let KB be a Horn knowledge base and $\dot{-}$ a contraction operator that satisfies all the rationality postulates for contraction $(\dot{-}1)$ to $(\dot{-}8)$. Then, there exists an order $\leq_{KB}$ among $\mathcal{S}$, that satisfies(≤ 1) to (≤ 5), and for any sentence α, $Mod(KB\dot{-}\alpha)$ is given as $Mod(KB) \cup Min(Mod(\{\neg\alpha\} \cup IC), \leq_{KB})$.*

Proof. *Follows from Theorem 18 and Theorem 19.*

Theorem 20. *Let KB be contracted by α, and $KB\dot{-}\alpha$ be obtained by the construction discussed above. Then $\dot{-}$ is a contraction operator iff it satisfies all the rationality postulates $(\dot{-}1)$ to $(\dot{-}8)$.*

Proof. *Follows from Lemma 10 and Lemma 11*

10.3 Relationship with the coherence approach of AGM

Given Horn knowledge base $KB = < P, Ab, IC, K >$ represents a belief set $KB^\bullet$ that is closed under Cn_P. We have defined how KB can be expanded, revised, or contracted. The question now is: *does our foundational approach (with respect to classical first-order logic) on KB coincide with coherence approach (with respect to our consequence operator Cn_P) of AGM on $KB^\bullet$?* There is a problem in answering this question (similar practical problem Baral & Zhang 2005) , since our approach, we require IC to be immutable, and only the current knowledge K is allowed to change. On the contrary, AGM approach treat every sentence in $KB^\bullet$ equally, and can throw out sentences from $Cn_P(IC)$. One way to solve this problem is to assume that sentences in $Cn_P(IC)$ are more entrenched than others. However, one-to-one correspondence can be established, when IC is empty. The key is our consequence operator Cn_P, and in the following, we show that coherence approach of AGM with this consequence operator, is exactly same as our foundational approach, when IC is empty.

Expansion

Expansion in AGM (Alchourron et al. 1985b) framework is defined as $KB\#\alpha = Cn_P(KB^\bullet \cup \{\alpha\})$, is is easy to see that this is equivalent to our definition of expansion (when IC is empty), and is formalized below.

Theorem 21. *Let $KB + \alpha$ be an expansion of KB by α. Then $(KB + \alpha)^\bullet = KB\#\alpha$.*

Proof. By our definition of expansion, $(KB+\alpha)^\bullet = Cn_P(IC \cup K \cup \{\alpha\})$, which is clearly the same set as $Cn_P(KB^\bullet \cup \{\alpha\})$.

Revision

AGM puts forward rationality postulates $(*1)$ to $(*8)$ to be satisfied by a revision operator on $KB^\bullet$. reproduced below:

(*1) (*Closure)* $KB^\bullet * \alpha$ is a belief set.
(*2) (*Success)* $\alpha \in KB^\bullet * \alpha$.
*(*3) (Expansion 1)* $KB^\bullet * \alpha \subseteq KB^\bullet\#\alpha$.
*(*4) (Expansion 2)* If $\neg\alpha \notin KB^\bullet$, then $KB^\bullet\#\alpha \subseteq KB^\bullet * \alpha$.
(*5) (*Consistency)*$KB^\bullet * \alpha$ is inconsistent iff $\vdash \neg\alpha$.
(*6) (*Preservation)* If $\vdash \alpha \leftrightarrow \beta$, then $KB^\bullet * \alpha = KB^\bullet * \beta$.
(*7) (*Conjunction 1)* $KB^\bullet * (\alpha \wedge \beta) \subseteq (KB^\bullet * \alpha)\#\beta$.
*(*8) (Conjunction 2)* If $\neg\beta \notin KB^\bullet * \alpha$, then,$(KB^\bullet * \alpha)\#\beta \subseteq KB^\bullet * (\alpha \wedge \beta)$.

The equivalence between our approach and AGM approach is brought out by the following two theorems.

Theorem 22. *Let KB a Horn knowledge base with an empty IC and $\dotplus$ be a revision function that satisfies all the rationality postulates $(\dotplus 1)$ to $(\dotplus 7)$. Let a revision operator $*$ on $KB^\bullet$ be defined as: for any sentence α, $KB^\bullet * \alpha = (KB \dotplus \alpha)^\bullet$. The revision operator *, thus defined satisfies all the AGM-postulates for revision $(*1)$ to $(*8)$.*

Proof.

*(*1) $KB^\bullet * \alpha$ is a belief set.*
This follows immediately, because $(KB \dotplus \alpha)^\bullet$ is closed with respect to Cn_P.
*(*2) $\alpha \in KB^\bullet * \alpha$.*
This follows from the fact that $\dotplus$ satisfies $(\dotplus 2)$.
*(*3) $KB^\bullet * \alpha \subseteq KB^\bullet\#\alpha$.*

*(*4) If $\neg\alpha \notin KB^\bullet$, then $KB^\bullet\#\alpha \subseteq KB^\bullet * \alpha$.*
These two postulates follow from $(\dotplus 4)$ and theorem 21.
*(*5) $KB^\bullet * \alpha$ is inconsistent iff $\vdash \neg\alpha$.*
This follows from from $(\dotplus 3)$ and our assumption that IC is empty.
*(*6) If $\vdash \alpha \leftrightarrow \beta$, then $KB^\bullet * \alpha = KB^\bullet * \beta$.*
This corresponds to $(\dotplus 5)$.
*(*7) $KB^\bullet * (\alpha \wedge \beta) \subseteq (KB^\bullet * \alpha)\#\beta$. This follows from $(\dotplus 6)$ and theorem 21.*

*(*8)* *If* $\neg\beta \notin KB^{\bullet} * \alpha$, *then,*$(KB^{\bullet} * \alpha)\#\beta \subseteq KB^{\bullet} * (\alpha \wedge \beta)$.
This follows from $(\dotplus 7)$ *and theorem 21*

Theorem 23. *Let* KB *a Horn knowledge base with an empty* IC *and* * *a revision operator that satisfies all the* AGM*-postulates* $(*1)$ *to* $(*8)$*. Let a revision function* $+$ *on* KB *be defined as: for any sentence* α, $(KB \dotplus \alpha)^{\bullet} = KB^{\bullet} * \alpha$. *The revision function* $+$, *thus defined, satisfies all the rationality postulates* $(\dotplus 1)$ *to* $(\dotplus 7)$.

Proof.

$(\dotplus 1)$ *P, Ab and* IC *do not change.*
Obvious.
$(\dotplus 2)$ α *is accepted in* $KB \dotplus \alpha$.
Follows from $(^*2)$.
$(\dotplus 3)$ *If* α *is satisfiable and consistent with* IC, *then* $KB \dotplus \alpha$ *is consistent.*
Since we have assumed IC *to be empty, this directly corresponds to* $(^*5)$.
$(\dotplus 4)$ *If* $\neg\alpha$ *is not accepted in* KB, *then* $KB \dotplus \alpha \equiv KB + \alpha$.
Follows from $(^*3)$ *and* $(^*4)$.
$(+5)$ *If* $KB \equiv KB'$ *and* $\alpha \equiv \beta$, *then* $KB \dotplus \alpha \equiv KB' \dotplus \beta$.
Since $KB \equiv KB'$ *they represent same belief set, i.e.* $KB^{\bullet} = KB'^{\bullet}$. *Now, this postulate follows immediately from* $(^*6)$.
$(\dotplus 6)$ $(KB \dotplus \alpha) + \beta$ *implies* $KB \dotplus (\alpha \wedge \beta)$.
Corresponds to $(^*7)$.
$(\dotplus 7)$ *If* $\neg\beta$ *is not accepted in* $KB \dotplus \alpha$, *then* $KB \dotplus (\alpha \wedge \beta)$ *implies* $(KB \dotplus \alpha) + \beta$.
Corresponds to $(^*8)$.

Contraction

AGM puts forward rationality postulates (-1) to (-8) to be satisfied by a contraction operator on closed set $KB^{\bullet}$, reproduced below:

(-1) (*Closure)* $KB^{\bullet} - \alpha$ is a belief set.
(-2) (*Inclusion)* $KB^{\bullet} - \alpha \subseteq KB^{\bullet}$.
(-3) (*Vacuity)* If $\alpha \notin KB^{\bullet}$, then $KB^{\bullet} - \alpha = KB^{\bullet}$.
(-4) (*Success)* If $\nvdash \alpha$, then $\alpha \notin KB^{\bullet} - \alpha$.
(-5) (*Preservation)* If $\vdash \alpha \leftrightarrow \beta$, then $KB^{\bullet} - \alpha = KB^{\bullet} - \beta$.
(-6) (*Recovery)* $KB^{\bullet} \subset (KB^{\bullet} - \alpha) + \alpha$.
(-7) (*Conjunction 1)*$KB^{\bullet} - \alpha \cap KB^{\bullet} - \beta \subseteq KB^{\bullet} - (\alpha \wedge \beta)$.
(-8) (*Conjunction 2)* If $\alpha \notin KB^{\bullet} - (\alpha \wedge \beta)$, then $KB^{\bullet} - (\alpha \wedge \beta) \subseteq KB^{\bullet} - \alpha$.

As in the case of revision, the equivalence is brought out by the following theorems. Since contraction is constructed in terms of revision, these theorems are trivial.

Corollary 1. *Let* KB *be a Horn knowledge base with an empty* IC *and* $\dot{-}$ *be a contraction function that satisfies all the rationality postulates* $(\dot{-}1)$ *to* $(\dot{-}8)$. *Let a contraction operator* $-$ *on* $KB^{\bullet}$ *be defined as: for any sentence* α, $KB^{\bullet} - \alpha = (KB \dot{-} \alpha)^{\bullet}$. *The contraction operator* $-$, *thus defined, satisfies all the* AGM *-postulates for contraction* (-1) *to* (-8).

Proof. Follows from Theorem 18 and Theorem 22

Corollary 2. *Let KB be a Horn knowledge base with an empty IC and $-$ be a contraction operator that satisfies all the AGM- postulates (-1) to (-8). Let a contraction function $\dot{-}$ on KB be defined as: for any sentence α, $(KB\dot{-}\alpha)^{\bullet} = KB^{\bullet} - \alpha$. The contraction function $\dot{-}$, thus defined, satisfies all the rationality postulates $(\dot{-}1)$ to $(\dot{-}8)$.*

Proof. Follows from Theorem 19 and Theorem 23

10.4 Realizing Horn knowledge base dynamics using abductive explanations

In this section, we explore how belief dynamics can be realized in practice (see (Aravindan & Dung 1994), (Aravindan 1995) and (Bessant et al. 1998)). Here, we will see how revision can be implemented based on the construction using models of revising sentence and an order among them. The notion of abduction proves to be useful and is explained in the sequel.

Let α be a sentence in $\mathcal{L}$. An *abductive explanation* for α with respect to KB is a set of abductive literals [5] Δ s.t. Δ consistent with IC and $\Delta \models_P \alpha$ (that is $\alpha \in Cn_P(\Delta)$). Further Δ is said to be *minimal* iff no proper subset of Δ is an abductive explanation for α.

The basic idea to implement revision of a Horn knowledge base KB by a sentence α, is to realize $Mod(\{\alpha\} \cup IC)$ in terms of abductive explanations for α with respect to KB. We first provide a useful lemma.

Definition 49. *Let KB be a Horn knowledge base, α a sentence, and Δ_1 and Δ_2 be two minimal abductive explanations for α with respect to KB. Then, the disjunction* of Δ_1 and Δ_2, written as $\Delta_1 \vee \Delta_2$, is given as:

$$\Delta_1 \vee \Delta_2 = (\Delta_1 \cap \Delta_2) \cup \{\alpha \vee \beta | \alpha \in \Delta_1 \backslash \Delta_2 \text{ and } \beta \in \Delta_2 \backslash \Delta_1\}.$$

Extending this to $\Delta^{\bullet}$, a set of minimal abductive explanations for α with respect to KB, $\vee\Delta^{\bullet}$ is given by the disjunction of all elements of $\Delta^{\bullet}$.

Lemma 12. *Let KB be a Horn knowledge base, α a sentence,and Δ_1 and Δ_2 be two minimal abductive explanations for α with respect to KB. Then,$Mod(\Delta_1 \vee \Delta_2) = Mod(\Delta_1) \cup Mod(\Delta_2)$.*

Proof. First we show that every model of Δ_1 is a model of $\Delta_1 \vee \Delta_2$. Clearly, a model M of Δ_1 satisfies all the sentences in $(\Delta_1 \cap \Delta_2)$. The other sentences in $(\Delta_1 \vee \Delta_2)$ are of the form $\alpha \vee \beta$, where α is from Δ_1 and β is from Δ_2. Since M is a model of Δ_1, α is true in M, and hence all such sentences are satisfied by M. Hence M is a model of $\Delta_1 \vee \Delta_2$ too. Similarly, it can be shown that every model of Δ_2 is a model of $\Delta_1 \vee \Delta_2$ too.

[5] An abductive literal is either an abducible A from Ab, or its negation $\neg A$.

Now, it remains to be shown that every model M of $\Delta_1 \vee \Delta_2$ is either a model of Δ_1 or a model of Δ_2. We will now show that if M is not a model of Δ_2, then it must be a model of Δ_1. Since M satisfies all the sentences in $(\Delta_1 \cap \Delta_2)$, we need only to show that M also satisfies all the sentences in $\Delta_1 \backslash \Delta_2$. For every element $\alpha \in \Delta_1 \backslash \Delta_2$: there exists a subset of $(\Delta_1 \vee \Delta_2)$, $\{\alpha \vee \beta | \beta \in \Delta_2 \backslash \Delta_2\}$. M satisfies all the sentences in this subset. Suppose M does not satisfy α, then it must satisfy all $\beta \in \Delta_1 \backslash \Delta_2$. This implies that M is a model of Δ_2, which is a contradictory to our assumption. Hence M must satisfy α, and thus a model Δ_1. Similarly, it can be shown that M must be a model of Δ_2 if it is not a model of Δ_1.

As one would expect, all the models of revising sentence α can be realized in terms abductive explanations for α, and the relationship is precisely stated below.

Lemma 13. *Let KB be a Horn knowledge base, α a sentence, and $\Delta^\bullet$ the set of all minimal abductive explanations for α with respect to KB. Then $Mod(\{\alpha\} \cup IC) = Mod(\vee \Delta^\bullet)$.*

Proof. It can be easily verified that every model M of a minimal abductive explanation is also a model of α. Since every minimal abductive explanation satisfies IC, M is a model of $\alpha \cup IC$. It remains to be shown that every model M of $\{\alpha\} \cup IC$ is a model of one of the minimal abductive explanations for α with respect to KB. This can be verified by observing that a minimal abductive explanation for α with respect to KB can be obtained from M.

Thus, we have a way to generate all the models of $\{\alpha\} \cup IC$, and we just need to select a subset of this based on an order that satisfies (≤ 1) to (≤ 5). Suppose we have such an order that satisfies all the required postulates, then this order can be mapped to a particular set of abductive explanations for α with respect to KB. This is stated precisely in the following theorem. An important implication of this theorem is that there is no need to compute all the abductive explanations for α with respect to KB. However, it does not say which abductive explanations need to be computed.

Theorem 24. *Let KB be a Horn knowledge base, and $\leq_{KB}$ be an order among abductive interpretations in $\mathcal{S}$ that satisfies all the rationality axioms (≤ 1) to (≤ 5). Then, for every sentence α, there exists $\Delta^\bullet$ a set of minimal abductive explanations for α with respect to KB, s.t. $Min(Mod(\{\alpha\} \cup IC), \leq_{KB})$ is a subset of $Mod(\vee \Delta^\bullet)$, and this does not hold for any proper subset of $\Delta^\bullet$.*

Proof. From Lemma 12 and Lemma 13, it is clear that $Mod(\{\alpha\} \cup IC)$ is the union of all the models of all minimal abductive explanations of α with respect to KB. Min selects a subset of this, and the theorem follows immediately. .

The above theorem 24, is still not very useful in realizing revision. We need to have an order among all the interpretations that satisfies all the required axioms, and need to compute all the abductive explanations for α with respect to KB.

The need to compute all abductive explanations arises from the fact that the converse of the above theorem does not hold in general. This scheme requires an universal order $\leq$, in the sense that same order can be used for any Horn knowledge base. Otherwise, it would be necessary to specify the new order to be used for further modifying $(KB \dotplus \alpha)$. However, even if the order can be worked out, it is not desirable to demand all abductive explanations of α with respect to KB be computed. So, it is desirable to work out, when the converse of the above theorem is true. The following theorem says that, suppose α is rejected in KB, then revision of KB by α can be worked out in terms of some abductive explanations for α with respect to KB.

Theorem 25. *Let KB be a Horn knowledge base, and a revision function $\dotplus$ be defined as: for any sentence α that is rejected in KB, $Mod(KB \dotplus \alpha)$ is a non-empty subset of $Mod(\vee\Delta^{\bullet})$, where $\Delta^{\bullet}$ is a set of all minimal abductive explanations for α with respect to KB. Then, there exists an order $\leq_{KB}$ among abductive interpretations in $\mathcal{S}$, s.t. $\leq_{KB}$ satisfies all the rationality axioms (≤ 1) to (≤ 5) and $Mod(KB \dotplus \alpha) = Min(Mod(\{\alpha\} \cup IC), \leq_{KB})$.*

Proof. Let us construct an order $\leq_{KB}$ among interpretations in $\mathcal{S}$ as follows: For any two abductive interpretations I and I' in $\mathcal{S}$, define $I \leq_{KB} I'$ iff either $I \in Mod(\vee\Delta^{\bullet})$ or $I \in Mod(KB \dotplus form(I, I'))$, where $form(I, I')$ stands for sentence whose only models are I and I'. We will show that $\leq_{KB}$ thus constructed satisfies (≤ 1) to (≤ 5) and $Min(Mod(\{\alpha\} \cup IC), \leq_{KB}) = Mod(KB \dotplus \alpha)$.

First, we show that $Min(Mod(\{\alpha\} \cup IC), \leq_{KB}) = Mod(KB \dotplus \alpha)$.Suppose α is not satisfiable, i.e. $Mod(\alpha)$ is empty, or α does not satisfy IC, then there are no abductive models of $\{\alpha\} \cup IC$, and hence $Min(Mod(\{\alpha\} \cup IC), \leq_{KB})$ is empty. From $(\dotplus 3)$, we infer that $Mod(KB \dotplus \alpha)$ is also empty. When α is satisfiable and α satisfies IC, the required result is obtained in two parts:

1) *If $I \in Min(Mod(\{\alpha\} \cup IC), \leq_{KB})$, then $I \in Mod(KB \dotplus \alpha)$*
   Since α is satisfiable and consistent with IC, $(\dotplus 3)$ implies that there exists at least one model, say I', for $KB \dotplus \alpha$. From $(\dotplus 1)$, it is clear that I' is a model of IC, from $(\dotplus 2)$ we also get that I' is a model of α, and consequently $I \leq_{KB} I'$ (because $I \in Min(Mod(\{\alpha\} \cup IC), \leq_{KB})$). Suppose $I \in (Mod(\vee\Delta^{\bullet}))$, then $(\dotplus 4)$ immediately gives $I \in Mod(KB \dotplus \alpha)$. If not, from our definition of $\leq_{KB}$, it is clear that $I \in Mod(KB \dotplus form(I, I'))$. Note that $\alpha \wedge form(I, I') \equiv form(I, I')$, since both I and I' are models of α. From $(\dotplus 6)$ and $(\dotplus 7)$, we get $Mod(KB \dotplus \alpha) \cap \{I, I'\} = Mod(KB \dotplus form(I, I'))$. Since $I \in Mod(KB \dotplus form(I, I'))$, it immediately follows that $I \in Mod(KB \dotplus \alpha)$.
2) *If $I \in Mod(KB \dotplus \alpha)$, then $I \in Min(Mod(\{\alpha\} \cup IC), \leq_{KB})$.*
   From $(\dotplus 1)$ we get I is a model of IC, and from $(\dotplus 2)$, we obtain $I \in Mod(\alpha)$. Suppose $I \in (Mod(\vee\Delta^{\bullet}))$, then from our definition of $\leq_{KB}$, we get $I \leq_{KB} I'$, for any other model I' of α and IC, and hence $I \in Min(Mod(\{\alpha\} \cup IC), \leq_{KB})$. Instead, if I is not a model of KB, then, to get the required result, we should show that $I \in Mod(KB \dotplus form(I, I'))$, for every model I' of α and IC. As we have observed previously, from $(\dotplus 6)$ and $(\dotplus 7)$, we get $Mod(KB \dotplus \alpha) \cap \{I, I'\} = Mod(KB \dotplus form(I, I'))$. Since $I \in Mod(KB \dotplus \alpha)$,

it immediately follows that $I \in Mod(KB \dotplus form(I, I'))$. Hence $I \leq_{KB} I'$ for any model I' of α and IC, and consequently, $I \in Min(Mod(\{\alpha\} \cup IC), \leq_{KB}$).

Every model of $Mod(KB \dotplus \alpha)$ is strictly minimal than all other interpretations. It is easy to verify that such a pre-order satisfies (≤ 1) to (≤ 5). In particular, since α is rejected in KB, (≤ 3) faithfulness is satisfied, and since non-empty subset of $Mod(\vee \Delta^{\bullet})$ is selected, (≤ 4) is also satisfied.

An important corollary of this theorem is that, revision of KB by α can be realized just by computing *one* abductive explanation of α with respect to KB, and is stated below.

Corollary 3. *Let KB be a Horn knowledge base, and a revision function $\dotplus$ be defined as: for any sentence α that is rejected in KB, $Mod(KB \dotplus \alpha)$ is a non-empty subset of $Mod(\Delta)$, where Δ is an abductive explanations for α with respect to KB. Then, there exists an order $\leq_{KB}$ among abductive interpretations in $\mathcal{S}$, s.t. $\leq_{KB}$ satisfies all the rationality axioms (≤ 1) to (≤ 5) and $Mod(KB \dotplus \alpha) = Min(Mod(\{\alpha\} \cup IC), \leq_{KB})$.*

The precondition that α is rejected in KB is not a serious limitation in various applications such as database updates and diagnosis, where close world assumption is employed to infer negative information. For example, in diagnosis it is generally assumed that all components are functioning normally, unless otherwise there is specific information against it. Hence, a Horn knowledge base in diagnosis either accepts or rejects normality of a component, and there is no ”don’t know” third state. In other words, in these applications the Horn knowledge base is assumed to be complete. Hence, when such a complete Horn knowledge base is revised by α, either α is already accepted in KB or rejected in KB, and so the above scheme works fine.

11 Related Works

We begin by recalling previous work on view deletion. Aravindan (Aravindan & Dung 1994), (Aravindan 1995), defines a contraction operator in view deletion with respect to a set of formulae or sentences using Hansson’s (Hansson 1997a) belief change. Similar to our approach (Delhibabu & Lakemeyer 2013, Delhibabu & Behrend, 2014, Delhibabu 2014a, Delhibabu 2014b) he focused on set of formulae or sentences in knowledge base revision for view update with respect to insertion and deletion and formulae are considered at the same level. Aravindan proposed different ways to change knowledge base via only database deletion, devising particular postulate which is shown to be necessary and sufficient for such an update process.

Our Horn knowledge base consists of two parts, immutable part and updatable part, but our focus is on minimal change computations. The related works are, Eiter (Eiter & Makino 2007), Langlois (Langlois et al. 2008) and Delgrande

(Delgrande & Peppas 2011) are focus on Horn revision with different perspectives like prime implication, logical closure and belief level. Segerberg (Segerberg 1998) defined a new modeling technique for belief revision in terms of irrevocability on prioritized revision. Hansson constructed five types of non-prioritized belief revision. Makinson (Makinson 1997) developed dialogue form of revision AGM. Papini (Papini 2000) defined a new version of knowledge base revision. In this paper, we considered the immutable part as a Horn clause (Fermé & Hansson 2001 shown shielded contraction similar to immutable part, the success postulate does not hold in general; some non-tautological beliefs are shielded from contraction and cannot be given up. Shielded contraction has close connections with credibility limited revision shown Hansson et al 2001) and the updatable part as an atom (literal). Knowledge bases have a set of integrity constraints.

Hansson's (Hansson 1997a) kernel change is related to abductive method. Aliseda's (Aliseda 2006) book on abductive reasoning is one of the motivation step. Christiansen's (Christiansen & Dahl 2009) work on dynamics of abductive logic grammars exactly fits our minimal change (insertion and deletion). Wrobel's (Wrobel 1995) definition of first order theory revision was helpful to frame our algorithm.

On the other hand, we are dealing with view update problem. Keller's (Keller 1985) thesis is motivation of the view update problem. There are many papers related to the view update problem (for example, the recent survey paper on view update by Chen and Liao (Chen & Liao 2010) and the survey paper on view update algorithms by Mayol and Teniente (Mayol & Teniente 1999). More similar to our work is the paper presented by Bessant (Bessant et al. 1998), which introduces a local search-based heuristic technique that empirically proves to be often viable, even in the context of very large propositional applications. Laurent (Laurent et al. 1998), considers updates in a deductive database in which every insertion or deletion of a fact can be performed in a deterministic way.

Furthermore, and at a first sight more related to our work, some work has been done on ”core-retainment” (Hansson 1991) in the model of language splitting introduced by Parikh (Parikh 1999). More recently, Doukari (Doukari et al. 2008), Özçep (Özçep 2012) and Wu (Wu et al. 2011) applied similar ideas for dealing with knowledge base dynamics. These works represent motivation step for our future work. Second, we are dealing with how to change minimally in the theory of ”principle of minimal change”, but current focus is on finding second best abductive explanation (Liberatore & Schaerf 2004 and 2012), 2-valued minimal hypothesis for each normal program (Pinto & Pereira 2011). Our work reflected in the current trends on Ontology systems and description logics (Qi and Yang (Qi & Yang 2008) and Kogalovsky (Kogalovsky 2012)). Finally, when we presented Horn knowledge base change in abduction framework, we did not talk about compilability and complexity (see the works of Liberatore (Liberatore 1997) and Zanuttini (Zanuttini 2003)).

The significance of our work can be summarized in the following:

- To define a new kind of revision operator on Horn knowledge base and obtain axiomatic characterization for it.

- To propose new generalized revision algorithm for Horn knowledge base dynamics, and study its connections with kernel change and abduction procedure.
- To develop a new view insertion algorithm for databases.
- To design a new view update algorithm for stratifiable Deductive Database (DDB), using an axiomatic method based on Hyper tableaux and magic sets.
- To study an abductive framework for Horn knowledge base dynamics.
- To present a comparative study of view update algorithms and integrity constraint.
- Finally, to shown connection between belief update versus database update.

12 Conclusion and remarks

The main contribution of this research is to provide a link between theory of belief dynamics and concrete applications such as view updates in databases. We argued for generalization of belief dynamics theory in two respects: to handle certain part of knowledge as immutable; and dropping the requirement that belief state be deductively closed. The intended generalization was achieved by introducing the concept of Horn knowledge base dynamics and generalized revision for the same. Further, we studied the relationship between Horn knowledge base dynamics and abduction resulting in a generalized algorithm for revision based on abductive procedures. The successfully demonstrated how Horn knowledge base dynamics provide an axiomatic characterization for update an literals to a stratifiable (definite) deductive database.

In bridging the gap between belief dynamics and view updates, we observe that a balance has to be achieved between computational efficiency and rationality. While rationally attractive notions of generalized revision prove to be computationally inefficient, the rationality behind efficient algorithms based on incomplete trees is not clear at all. From the belief dynamics point of view, we may have to sacrifice some postulates, vacuity, to gain computational efficiency. Further weakening of relevance has to be explored, to provide declarative semantics for algorithms based on incomplete trees.

On the other hand, from the database side, we should explore various ways of optimizing the algorithms that would comply with the proposed declarative semantics. We believe that partial deduction and loop detection techniques, will play an important role in optimizing algorithms. Note that, loop detection could be carried out during partial deduction, and complete SLD-trees can be effectively constructed wrt a partial deduction (with loop check) of a database, rather than wrt database itself. Moreover, we would anyway need a partial deduction for optimization of query evaluation.

We have presented two variants of an algorithm for update a view atom from a definite database. The key idea of this approach is to transform the given database into a logic program in such a way that updates can be read off from the models of this transformed program. We have also shown that this algorithm is rational in the sense that it satisfies the rationality postulates that are justified from philosophical angle. In the second variant, where materialized view is used for the transformation, after generating a hitting set and removing corresponding EDB atoms, we easily move to the new materialized view. An obvious way is to recompute the view from scratch using the new EDB (i.e., compute the Least Herbrand Model of the new updated database from scratch) but it is certainly interesting to look for more efficient methods.

Though we have discussed only about view updates, we believe that Horn knowledge base dynamics can also be applied to other applications such as view maintenance, diagnosis, and we plan to explore it further (see works (Biskup 2012) and (Caroprese et al. 2012)). Still, a lot of developments are possible, for improving existing operators or for defining new classes of change operators. In the relation of Horn KB revision with hitting set as to be describe similar construction for description logic. In particular assuming the T-Box to the hitting set and the rest with the A-Box (Delgrande, JP & Wassermann 2013). We did not talk about complexity (see the works of Liberatore ((Liberatore 1997) and (Liberatore & Schaerf 2004)), Caroprese (Caroprese 2012), Calvanese's (Calvanese 2012), and Cong (Cong et al. 2012)). In this thesis answer impotent question for experimental people that is,*any real life application for AGM in 25 year theory?* (Ferme & Hansson 2011). The revision and update are more challenging in logical view update problem (database theory), we extended the theory to combine our results similar in the Konieczny's (Konieczny 2011) and Nayak's (Nayak 2011).

Acknowledgement

I would like to thanks Chandrabose Aravindan and Gerhard Lakemeyer both my Indian and Germany PhD supervisor, give encourage to write the paper. I owe my deepest gratitude to Ramaswamy Ramanujam from Institute of Mathematical Sciences and Ulrich Furbach from University Koblenz-Landau for my PhD thesis member. I thank to my PhD thesis examiner's Eduardo Fermé from University of Madeira, Mohua Banerjee from Indian Institute of Technology Kanpur and Arindama Singh form Indian Institute of Technology Madras. The author acknowledges the support of SSN College of Engineering research funds and RWTH Aachen, where he is visiting scholar with an Erasmus Mundus External Cooperation Window India4EU by the European Commission.

Appendix A Tab. 1. Summary of view-update and integrity constraint with our axiomatic method

Method	Problem				Database schema				Update req.		Mechanism			Update Change			Solutions		
	Type	View Update	IC Enforce.	Run/ Comp.	Def. Lang.	View	IC def.	Kind of IC	Mul.	Update Operat.	Tech- nique	Base Facts	User Part.	Type	Base facts	View def.	Axiom	Sound.	Complete.
[69]	N	Yes	Check	Run	Logic	Yes	Yes	Static	No	$\iota\ \delta$	SLDNF	No	No	S	Yes	No	1-6,9	No	Not proved
[86]	N	Yes	Maintain	Run	Logic	Yes	Yes	Static	Yes	$\iota\ \delta$	SLDNF	No	No	S	Yes	No	—	No	No
[96]	S	Yes	Check	Run	Logic	Yes	Yes	Static	Yes	$\iota\ \delta$	—	Yes	No	SS	Yes	Yes	1-6,7	Not proved	Not proved
[118]	N	No	Maintain	Run	Logic	Yes	Yes	Static	Yes	$\iota\ \delta$	—	Yes	No	S	Yes	No	1-6,7	No proved	No proved
[70]	S	Yes	Check Maintain	Comp. Run	Relation. Logic	No	No	Static	Yes	$\iota\ \delta\ \chi$	predef. Programs	Yes	No	G	Yes	No	—	No	No
[51]	N	Yes	Check	Run	Logic	Yes	No	Static	Yes	$\iota\ \delta$	predef Programs	Yes	No	S	Yes	No	1-6,7	Not Proved	No
[146]	S	Yes	Check Maintain	Run	Logic	Yes	Yes	Static	Yes	$\iota\ \delta\ \chi$	SLDNF	No	No	SS	Yes	No	1-6,7	Yes	No
[71]	N	Yes	Maintain	Run	Logic	Yes	No	Static	Yes	$\iota\ \delta$	Unfold	Yes	No	SS	Yes	No	1-6,9	Yes	Yes
[114]	N	Yes	Maintain	Comp. Run	Logic	Yes	Yes	Static Dynamic	Yes	$\iota\ \delta\ \chi$	SLDNF	Yes	No	S	Yes	No	1-6,9	Not proved	Not proved
[150]	S	Yes	Maintain	Run	Logic	Yes	Yes	Static	Yes	$\iota\ \delta$	Unfold.	No	Yes	S	Yes	No	1-6,7	Not proved	No
[8]	H	Yes	Check	Run	Logic	Yes	Yes	Static	Yes	$\iota\ \delta$	SLD	Yes	No	S	Yes	No	1-6,9	Yes	Yes
[26]	N	No	Maintain	Comp Run	Relation Logic	Yes	Limited	Static	Yes	$\iota\ \delta\ \chi$	Active	Yes	Yes	S	Yes	No	—	No	No
[66]	N	No	Maintain	Comp Run	Relation Logic	No	Flat Limited	Static Dynamic	Yes	$\iota\ \delta\ \chi$	Active	Yes	Yes	S	Yes	No	—	No	No
[29]	H	Yes	Check Maintain	Comp. Run	O-O	Class Att.	Limited	Static	Yes	$\iota\ \delta$	Active	Yes	No	SS	Yes	No	1-6,9	No	Yes
[37]	N	Yes	Maintain	Run	Logic	Yes	Flat Limited	Static	Yes	$\iota\ \delta$	Unfold.	Yes	No	S	Yes	No	1-6,9	Not proved	Yes
[107]	N	Yes	Maintain	Run	Logic	Yes	Limited	Static	No	$\iota\ \delta$	Active	Yes	No	SS	Yes	No	1-6,7	Yes	Not proved
[145]	N	Yes	Maintain	Comp Run	Logic	Yes	Yes	Static Dynamic	Yes	$\iota\ \delta$	SLDNF	Yes	No	S	Yes	No	1-6,9	Yes	Yes
[138]	S	Yes	Maintain	Comp	Logic	No	Flat Limited	Static	Yes	$\iota\ \delta$	predef Programs	—	Yes	G	No	Yes	—	No	Not proved

Method	**Problem**				**Database schema**				**Update req.**		**Mechanism**			**Update Change**			**Solutions**		
	Type	View Update	IC Enforce.	Run/ Comp.	Def. Lang.	View	IC def.	Kind of IC	Mul.	Update Operat.	Tech-nique	Base Facts	User Part.	Type	Base facts	View def.	Axiom	Sound.	Complete.
[144]	N	No	Maintain	Comp	Logic	Yes	Limited	Static	Yes	$\iota\ \delta\ \chi$	Predef Program	Yes	No	S	Yes	No	1-6,7	Yes	No
[10]	H	Yes	Check	Run	Logic	Yes	Limited	Static	Yes	$\iota\ \delta$	SLD	Yes	No	S	Yes	No	1-6,9	Yes	Yes
[40]	N	Yes	Maintain	Run	Logic	Yes	Yes	Static	Yes	$\iota\ \delta$	SLDNF	No	No	S	Yes	No	1-6,7	No	Not Proved
[105]	N	Yes	Maintain	Run	Logic	Yes	Flat Limited	Static	Yes	$\iota\ \delta$	Unfold	No	Yes	G	Yes	No	1-6,7	Not proved	No
[151]	H	No	Maintain	Comp. Run	Relation	Yes	Limited	Static Dynamic	Yes	$\iota\ \delta\ \chi$	Unfold	Yes	No	S	Yes	No	1-6,7	Not proved	Not proved
[109]	N	No	Maintain Restore	Comp Run	Logic	No	Flat Limited	Static Dynamic	Yes	$\iota\ \delta$	Active	Yes	No	G	No	No	—	No	No
[139]	N	No	Maintain	Comp Run	Relation	No	Flat Limited	Static	Yes	$\iota\ \delta$	Active	Yes	No	S	No	No	—	No	No
[108]	N	Yes	Check	Run	Logic	Yes	Limited	Static	Yes	$\iota\ \delta$	SLD	Yes	No	S	Yes	No	1-6,9	Yes	Yes
[1]	O	No	Maintain	Run	Logic	Yes	Limited	Static	Yes	$\iota\ \delta$	—	Yes	No	S	Yes	No	—	No	No
[140]	N	No	Maintain	Comp	Relation	No	Limited	Static	Yes	$\iota\ \delta$	Predef Program	No	No	G	No	No	—	No	No
[72]	N	No	Maintain	Comp	Logic	Yes	Limited	Static	Yes	$\iota\ \delta$	—	No	No	S	No	No	—	No	No
[30]	N	No	Maintain	Comp. Run	Relation	Yes	Limited	Static Dynamic	Yes	$\iota\ \delta$	—	Yes	No	S	Yes	No	—	Not proved	Not proved
[52]	H	Yes	Check	Run	Logic	Yes	Yes	Static	Yes	$\iota\ \delta$	Predef Programs	Yes	No	S	Yes	No	1-6,7	Yes	Not proved
[78]	O	Yes	Check	Run	Relation	Yes	Limited	Static	No	$\iota\ \delta$	Unfold	Yes	No	S	Yes	No	1-6,9	Yes	Yes
[14]	O	Yes	Check	Run	Relation	Yes	Limited	Static	No	$\iota\ \delta$	Unfold	Yes	No	S	Yes	No	1-6,9	Yes	Yes
[56]	N	Yes	Check	Run	Logic	Yes	Yes	Static	Yes	$\iota\ \delta$	SLDNF	Yes	No	S	Yes	No	1-6,9	Yes	Yes
[132]	N	No	Maintain	Run	Logic	Yes	Limited	Static	Yes	$\iota\ \delta$	Predef Programs	Yes	No	S	Yes	No	1-6,7	Yes	Not proved
[81]	O	No	Maintain	Comp	Relation	Yes	Limited	Static	Yes	$\iota\ \delta$	—	Yes	No	S	Yes	No	—	No	No

Method	**Problem**				**Database schema**				**Update req.**		**Mechanism**			**Update Change**			**Solutions**		
	Type	View Update	IC Enforce.	Run/ Comp.	Def. Lang.	View	IC def.	Kind of IC	Mul.	Update Operat.	Tech- nique	Base Facts	User Part.	Type	Base facts	View def.	Axiom	Sound.	Complete.
[135]	N	Yes	Check	Run	Logic	No	Limited	Static	Yes	$\iota\ \delta$	SLDNF	Yes	No	S	Yes	No	1-6,9	Yes	Yes
[57]	N	Yes	Check	Run	Logic	Yes	Yes	Static	No	$\iota\ \delta$	SLDNF	No	No	S	Yes	No	1-6,9	Not	No proved
[113]	N	Yes	Check Maintain	Run	Logic	Yes	Yes	Static	No	$\iota\ \delta\ \chi$	SLD	Yes	No	S	Yes	No	—	No	No
[19]	N	Yes	Check Maintain	Run Comp	Logic	Yes	Yes	Static	No	$\iota\ \delta\ \chi$	SLD	Yes	No	SS	Yes	No	1-6,7	Yes	Not proved
[31]	N	Yes	Check Maintain	Run	Logic	Yes	Yes	Static Dynamic	Yes	$\iota\ \delta\ \chi$	Predef Program	Yes	No	S	Yes	No	1-6,9	Yes	Yes
[25]	N	Yes	Check	Comp	Logic	Yes	Yes	Dynamic	Yes	$\iota\ \delta$	Predef Programs	Yes	No	S	Yes	No	—	Not Proved	No proved
[32]	N	Yes	Check Maintain	Run	Logic	Yes	Yes	Static Dynamic	Yes	$\iota\ \delta\ \chi$	Predef Program	Yes	No	S	Yes	No	1-6,9	Yes	Yes
[35]	N	No	Maintain	Comp	Logic	Yes	No	—	Yes	$\iota\ \delta$	—	Yes	No	S	Yes	No	—	No	No
[154]	N	No	Maintain	Run	Relation	Yes	No	—	Yes	$\iota\ \delta\ \chi$	Unfold	Yes	No	SS	No	No	—	Not proved	Not proved
[79]	O	No	Maintain	Comp. Run	Logic	Yes	No	—	Yes	$\iota\ \delta$	—-	Yes	No	G	No	No	—	Yes	Not proved
[15]	S	Yes	Check	Run	Logic	Yes	Flat Limited	Static	Yes	$\iota\ \delta$	SLDNF	Yes	No	S	Yes	No	1-6,9	Yes	Not proved
[53]	N	Yes	Check	Run	Logic	Yes	Yes	Static	Yes	$\iota\ \delta$	—	Yes	No	S	Yes	No	—	No	No
[27]	O	No	Maintain	Run	Relation	Yes	No	Static	Yes	$\iota\ \delta$	SLD	Yes	Yes	G	No	No	—	Not proved	Not proved
[16]	O	No	Maintain	Comp	Relation	Yes	No	Static	Yes	$\iota\ \delta\ \chi$	—	Yes	No	SS	Yes	No	—	No	No
[4]	O	No	Maintain	Comp. Run	Relation	No	Limited	Static Dynamic	Yes	$\iota\ \delta$	—	Yes	No	G	Yes	No	—	No	No
[111]	N	No	Maintain	Comp	Relation	No	Yes	Static	Yes	$\iota\ \delta\ \chi$	Unfold	No	Yes	SS	No	No	—	No	No
[137]	N	No	Check	Comp	Logic	No	Yes	Static	Yes	$\iota\ \delta$	Active	Yes	No	G	Yes	No	—	No	No
[41]	N	Yes	Check	Run	Logic	Yes	Yes	Static	Yes	$\iota\ \delta$	SLD	Yes	No	S	Yes	No	1-6,9	Yes	Yes

References

1. Agrawal, S, Chaudhuri, S & Narasayya, VR 2008. 'Automated selection of materialized views and indexes in SQL databases' Proceedings of the 26th International Conference on Very Large Data Bases, ed. Abbadi, AE, Cairo, Egypt, pp. 496-505.
2. Alchourron, CE & Makinson, D 1985, 'On the logic of theory change: Safe contraction', Studia Logica, vol. 44, pp. 405-422.
3. Alchourron, CE, Gärdenfors & Makinson, D 1985, 'On the logic of theory change: Partial meet contraction and revision functions', Journal of Symbolic Logic 50, pp. 510 - 530.
4. Alexandre, L & Coelho, J 2011, 'Filtering XML content for publication and presentation on the web',Digital Information Management (ICDIM), pp. 85-89.
5. Aliseda, A 2006, 'Abductive Resoning Logic Investigations into Discovery and Explanation', Springer book series, vol. 330.
6. Aliseda, A 2007, 'Abductive Reasoning: Challenges Ahead', THEORIA, vol. 22, no. 3, pp. 261-270.
7. Apt, K.P & Bezem, M 1991, 'Acyclic Programs', New Generation Computing, vol. 9, no. 3, pp. 335-363
8. Aravindan, C & Dung, PM 1994, 'Belief Dynamics, Abduction, and Database', Logics in Artificial Intelligence Lecture Notes in Computer Science, ed. MacNish, C, York, UK, vol. 838, pp. 66-85.
9. Aravindan, C 1995, 'Dynamics of Belief: Epistmology, Abduction and Database Updat', Phd Thesis, AIT.
10. Aravindan, C & Baumgartner, P 1997, 'A Rational and Efficient Algorithm for View Deletion in Databases', Logic Programming proceedng International Symposium, pp. 165-179.
11. Arieli, O, Denecker, M, Nuffelen, BV & Bruynooghe, M 2004,'Coherent Integration of Databases by Abductive Logic Programming',Journal of Artificial Intelligence Research, vol. 21, pp. 245-286.
12. Baral, C & Zhang, Y 2005, 'Knowledge updates: Semantics and complexity issues', Atificial Intelligence, vol. 164, no. 1, pp. 209-243.
13. Baumgartner, P, Fröhlich, P, Furbach, U & Nejdl, W 1997, 'Semantically Guided Theorem Proving for Diagnosis Applications', Fifteenth International Joint Conference on. Artificial Intelligence, vol. 1, pp. 460-465.
14. Bauer, A & Lehne, W 2003, 'On solving the view selection problem in distributed data warehouse architectures', 15th International Conference on Scientific and Statistical Database Management, pp. 43-57.
15. Behrend, A & Manthey, R 2008, 'A Transformation-Based Approach to View Updating in Stratifiable Deductive Databases', Foundations of Information and Knowledge Systems Lecture Notes in Computer Science, ed. Hartmannp, S, vol. 4932, pp. 253-271.
16. Bellahsene, Z, Cart, M & Kadi, N 2010, 'A cooperative approach to view selection and placement in P2P systems', Proceedings of the 2010 international conference on On the move to meaningful internet systems, vol. 1, pp. 515-522.
17. Benferhat, S, Lagrue, S & Papini, O 2005, 'Revision of partially ordered information: Axiomatization, semantics and iteration', International Joint Conference on Artificial Intelligence, pp. 376-381.
18. Bessant, B, Gr*é*goire, E, Marquis, P & Sa*Ï*s, L 1998, 'Combining Nonmonotonic Reasoning and Belief Revision: A Practical Approach', Artificial Intelligence: Methodology, Systems, and Applications Lecture Notes in Computer Science, ed. Giunchiglia, F, Sozopol, Bulgaria, vol. 1480, pp 115-128.

19. Braga, D, Campi, A & Martinenghi, D 2006, 'Efficient Integrity Checking over XML Documents', Current Trends in Database Technology EDBT 2006 Lecture Notes in Computer Science, ed. Grust, T, Munich, Germany, vol. 4254, pp. 206-219.
20. Biskup, J 2012, 'Inference-usability confinement by maintaining inference-proof views of an information system', Journal International Journal of Computational Science and Engineering, vol. 7, no. 1, pp 17-37.
21. Booth, R & Meyer, T 2006, 'Admissible and restrained revision', Journal of Artificial Intelligence Research, vol. 26, pp. 127-151.
22. Boutilier, C & Beche, V 1995, 'Abduction as belief revision', Artificial Intelligence, vol. 77, no.1, pp. 43 - 94.
23. Caroprese, L, Trubitsyna, I, Truszczynski, M & Zumpano,E 2012, 'The View-Update Problem for Indefinite Databases', Logics in Artificial Intelligence Lecture Notes in Computer Science, ed. Herzig, A, Toulouse, France, vol. 7519, pp. 134-146
24. Calvanese, D, Giacomob, GD, Lenzerinib & M, Rosatib, R 2012, 'View-based query answering in Description Logics Semantics and complexity', Journal of Computer and System Sciences, vol. 78, no. 1, pp. 26-46.
25. Caroprese, L, Trubitsyna, I & Zumpano, E 2007, 'View Updating Through Active Integrity Constraints', Logic Programming Lecture Notes in Computer Science, ed. Dahl, V, Porto, Portugal, vol. 4670, pp. 430-431.
26. Ceri, S, Fraternali, P, Paraboschi, S & Tanca, L 1994, 'Automatic Generation of Production Rules for Integrity Maintenance', ACM Transactions on Database Systems, vol. 19, no. 3, pp. 367-422.
27. Chaves, LWF, Buchmann, E, Hueske, F & Böhm, K 2009, 'Towards materialized view selection for distributed databases', Proceedings of the 12th International Conference on Extending Database Technology: Advances in Database Technology, pp. 1088-1109.
28. Chen, H & Liao, H 2010, 'A Comparative Study of View Update Problem', Data Storage and Data Engineering, pp. 83-89.
29. Chen, IA, Hull, R & McLeod, D 1995, 'An Execution Model for Limited Ambiguity Rules and Its Application to Derived Data Update', ACM Transactions on Database Systems, vol. 20, no. 4, pp. 365-413.
30. Chirkova, R 2002, 'The view-selection problem has an exponential-time lower bound for conjunctive queries and views', ACM Symposium on Principles of Database Systems, pp. 159-168.
31. Christiansen, H & Martinenghi, D 2006, 'On Simplification of Database Integrity Constraints', Fundamenta Informaticae, vol. 71, no. 4, pp. 371-417.
32. Christiansen, H & Rekouts, M 2007, 'Integrity Checking and Maintenance with Active Rules in XML Databases', Databases BNCOD Workshops, pp. 59-67.
33. Christiansen, H & Dahl,V 2009, 'Abductive Logic Grammars', Logic, Language, Information and Computation Lecture Notes in Computer Science, ed. Ono, H, Tokyo, Japan, vol. 5514, pp. 170-181.
34. Codd, EF 1979, 'Extending the Database Relational Model to Capture More Meaning', ACM Transactions on Database Systems, vol. 4, no. 4, pp. 397-434.
35. Coelho, J & Florido, M 2007, 'Type-Based Static and Dynamic Website Verification', Internet and Web Applications and Services, pp. 32- 41.
36. Cong, G, Fan, W, Geerts, F, Li, J & Luo, J 2012, 'On the Complexity of View Update Analysis and Its Application to Annotation Propagation', Journal IEEE Transactions on Knowledge and Data Engineering, vol. 24, no. 3, pp. 506-519.
37. Console, L, Sapino, ML & Dupré, DT 1995, 'The Role of Abduction in Database View Updating', Journal of Intelligent Information Systems, vol. 4, pp. 261-280.

38. Dalal, M 1988, 'Investigations into a Theory of Base Revision: Preliminary report', In Seventh National Converence on Artificial Intelligence, (AAAI), St. Paul, pp:475-479.
39. Darwiche, A & Pearl, J 1997, 'On the logic of iterated belief revision', Artificial Intelligence, vol. 89, pp.1-29.
40. Decker, H 1997, 'One Abductive Logic Programming Procedure for two kind of Updates', Proceedings Workshop DINAMICS at Int. Logic Programming Symposium, pp. 23-36.
41. Delhibabu, R., & Lakemeyer, G. 2013, 'A Rational and Efficient Algorithm for View Revision in Databases', Applied Mathematics & Information Sciences, vol. 7, no. 3, pp. 843-856.
42. Delhibabu, R 2014, 'An Abductive Framework for Horn Knowledge Base Dynamics', Applied Mathematics & Information Sciences, vol 9, no. 2, pp. 571-582.
43. Delhibabu, R & Behrend, A 2014, 'A New Rational Algorithm for View Updating in Relational Databases', Applied Intelligence, Accepted.
44. Delhibabu, R 2014, 'Comparative Study of View Update Algorithms in Rational Choice Theory', Applied Intelligence - Accepted.
45. Delhibabu, R 2014, 'A Better Understanding of the Dynamics of Belief Update VS Database Update', Studia Logica - Submitted.
46. Delgrande, JP 2008, 'Horn Clause Belief Change: Contraction Functions', KR, pp. 156-165
47. Delgrande, JP & Peppas, P 2011, 'Revising Horn Theories', of the Twenty-Second international joint conference on Artificial Intelligence, vol. 2, pp. 839-844.
48. Delgrande, JP & Wassermann, R 2013, 'Horn Clause Contraction Functions', Journal of Artificial Intelligence Research, vol.48, pp: 475-511.
49. Denecker, M & Kakas, AC 2002, 'Abduction in Logic Programming', Computational Logic: Logic Programming and Beyond, pp. 402-436.
50. Dhote, C. A & Ali, MS 2009, 'Materialized View Selection in Data Warehousing: A Survey', Journal of Applied Sciences, pp. 401-414.
51. Dong, G & Topor, RW 1992, 'Incremental evaluation of datalog queries', In Database Theory - ICDT, ed. Biskup, J, Berlin, Germany, vol. 646, pp. 282-296.
52. Domínguez, E, Lloret, J & Zapataet, MA 2002, 'Integrity Constraint Enforcement by Means of Trigger Templates', Advances in Information Systems Lecture Notes in Computer Science, ed. DomÍınguez, E, Izmir, Turkey, vol. 2457, pp. 54-64.
53. Domínguez, E, Lloret, J, Rubio, AL & Zapata, MA 2008, 'Model-Driven, View-Based Evolution of Relational Databases', Database and Expert Systems Applications Lecture Notes in Computer Science, ed. Bhowmick, SB, Turin, Italy, vol. 5181, pp. 822-836.
54. Doukari, O, Jeansoulin, R & Würbelet, E 2008, 'A New Framework for Local Belief Revision', Advances in Artificial Intelligence Lecture Notes in Computer Science, vol. 5317, pp. 54-64.
55. Eiter, T & Makino,K 2007, 'On computing all abductive explanations from a propositional Horn theory', Journal of ACM, vol. 54, no. 5, pp: 1-54
56. Farré, C, Teniente, E & Urp*í*, T 2003, 'Handling Existential Derived Predicates in View Updating', Logic Programming Lecture Notes in Computer Science, vol. 2916, pp. 148-162.
57. Farré, C, Teniente, E & Urp*í*, T 2004, 'A New Approach for Checking Schema Validation Properties', Database and Expert Systems Applications Lecture Notes in Computer Science, ed. Galindo, F, Zaragoza, Spain, vol. 3180, pp. 77-86.

58. Falappa, MA, Kern-Isberner, G, Reis, MDL & Simari, GR 2012, 'Prioritized and Non-prioritized Multiple Change on Belief Bases', Journal of Philosophical Logic, vol. 41, no. 1, pp. 77-113.
59. Fermé , EL 1992, 'Actualizacion de Bases de Conocimiento usando Teorias de Cambio de Creenica', lberamia, pp 419-436.
60. Fermé , EL & Hansson, SO 2001, 'Shielded Contraction', Frontiers in Belief Revision, Applied Logic Series, Kluwer Academic Publishers, pp. 85-107.
61. Fermé , EL & Hansson, SO 2011, 'AGM 25 Years - Twenty-Five Years of Research in Belief Change', Journal of Philosophical Logic, vol. 40, no. 2, pp. 295-331.
62. Fraternali, P & Paraboschi, S 1993, 'A Review of Repairing Techniques for Integrity Maintenance', Rules in Database Systems Workshops in Computing, ed. Patonp, NW, Edinburgh, Scotland, pp. 333-346.
63. Gärdenfors, P 1992, 'Belief Revision', Cambridge University Press, Cambridge.
64. Gärdenfors, P 1998, 'Knowledge in flux', MIT Press, Cambridge.
65. Gärdenfors, P & Makinson, D 1988, 'Revisions of knowledge systems using epistemic entrenchment', Proceedings of the Second Conference on Theoretical Aspects of Reasoning about Knowledge, pp. 83-95.
66. Gertz, M 1994, 'Specifying Reactive Integrity Control for Active Databases', Research Issues in Data Engineering, Active Database Systems, pp. 62-70.
67. Godfrey, P, Grant, J, Gryz, J & Minker, J 1998, 'Integrity Constraints: Semantics and Applications', Logics for Databases and Information Systems, Kluwer Academic Publishers Norwell, MA, USA, pp. 265-306.
68. Grove, A 1988, 'Two modellings for theory change', Journal of Philosophical Logic, vol. 17, pp. 157-180.
69. Guessoum, A & Lloyd, JW 1990, 'Updating Knowledge Bases', New Generation Computing, vol. 8, no.1, pp. 71-89.
70. Gupta, A, Katiyar, D & Mumick, IS 1992, 'Counting solutions to the view maintenance problem', Proceedings of the Workshop on Deductive Databases held in conjunction with the Joint International Conference and Symposium on Logic Programming, pp. 185-194.
71. Gupta, A, Mumick, IS & Subrahmanian, VS 1993, 'Maintaining views incrementally', Proceedings of the 1993 ACM SIGMOD international conference on Management of data, vol. 157-176.
72. Halevy, AY 2001, 'Answering queries using views: A survey', Journal The VLDB Journal, vol. 10, no. 4, pp 270 - 174.
73. Hansson, SO 1991, 'Belief contraction without recovery', Studia Logica, vol. 50,no. 2, pp. 251-260.
74. Hansson, SO 1992, 'A dyadic representation of belief', Cambridge University Press, vol. 29, pp. 89-121.
75. Hansson, SO 1997, 'A Textbook of Belief Dynamics', Kluwer Academic Publishers, Dordrecht.
76. Hansson, SO 1997, 'Theoria', Special Issue on non-prioritized belief revision, Wiley, Chichester, vol. 63, no. 1.
77. Hansson, SO, Fermé, EL, Cantwell, J & Falappa M 2001, 'Credibility-Limited Revision', Journal of Symbolic Logic, vol. 66, no. 4, pp.1581-1596.
78. Hegner, SJ 2002, 'Uniqueness of Update Strategies for Database Views', Proceedings of the Second International Symposium on Foundations of Information and Knowledge Systems, pp. 230-249.
79. Hegner, SJ 2007, 'A Model of Database Components and their Interconnection Based upon Communicating Views', Proceeding of the 2008 conference on Information Modelling and Knowledge Bases XIX, pp. 79-99.

80. Herzig, A & Rifi,O 1999, 'Propositional Belief Base Update and Minimal Change', Artificial Intelligence, vol. 115, no. 1, pp. 107-138.
81. Horng, JT, Chang, JY & Liu, BJ 2003, 'Applying evolutionary algorithms to materialized view selection in a data warehouse', Soft Computing, vol. 7, no. 8, pp. 574-591.
82. Inoue, K & Sakama, C 2008, 'Comparing Abductive Theories', Proceedings of the 2008 conference on ECAI 2008: 18th European Conference on Artificial Intelligence, pp. 35-39.
83. Inoue, K & Sakama, C 2006, 'Abductive Equivalence in First-order Logic', Logic Journal of the IGPL, vol. 14, no. 2, vol. 333-346.
84. Jackson, EK & Schulte, W. (2008). Model Generation for Horn Logic with Stratified Negation. Proceedings of the 28th IFIP WG 6.1 international conference on Formal Techniques for Networked and Distributed Systems, pp. 1-20.
85. Jin, Y & Thielscher, M 2007, 'Iterated belief revision, revised', Artificial Intelligence, vol. 171, pp. 1-18.
86. Kakas, AC & Mancarella, P 1990, 'Database Updates Through Abduction', Proceedings of the 16th International Conference on Very Large Data Bases, pp. 650-661.
87. Katsuno, H & Mendelzon, AO. 1991, 'Propositional knowledge base revision and minimal change', Artificial Intelligence, vol. 52, pp. 263Ą0Č2294.
88. Katsuno, H & Mendelzon, AO 1992, 'On the difference between updating a knowledge base and revising it', Cambridge University Press, Cambridge.
89. Keller, A 1985, 'Updating Relational Databases Through Views', Stanford University, Phd Thesis.
90. Kogalovsky, MR 2012, 'Ontology-based data access systems', Programming and Computer Software, vol. 38, no. 4, pp. 167-182.
91. Konieczny, S, Lang, J & Marquis, P 2004, 'DA^2 merging operators', Artificial Intelligence, vol. 157, no. 1, pp. 49-79.
92. Konieczny, S & Pino P*é*rez, R 2008, 'Improvement operators', Proceedings Eleventh International Conference on Principles of Knowledge Representation and Reasoning, vol. 177-186.
93. Konieczny, S, Grespan, MM & P*é*rez, RP 2010, 'Taxonomy of improvement operators and the problem of minimal change', Proceedings of the Twelfth International Conference on the Principles of Knowledge Representation and Reasoning, pp. 161-170.
94. Konieczny, S 2011, 'Dynamics of Beliefs', Scalable Uncertainty Management Lecture Notes in Computer Science, vol. 6929, pp. 61-74.
95. Kowalski, R 1994, 'Logic without model theory', Technical Report, Department of Computing, Imperial College, London, U.K.
96. Küchenhoff 1991, 'On the efficient computation of the difference between consecutive database states', Deductive and Object-Oriented Databases Lecture Notes in Computer Science, ed. Delobel, C, Munich, Germany, vol. 566, pp. 478-502.
97. Lakemeyer, G 1995, 'A Logical Account of Relevance', International Joint Conference on AI, vol. 1, pp. 53-861.
98. Langlois, M, Sloan, RH, Sz*ö*r*é*nyi, B & Thr*á*n, G 2008, 'Horn Complements: Towards Horn-to-Horn Belief Revision', Proceedings of the 23rd national conference on Artificial intelligence, vol. 1, pp. 466-471.
99. Laurent, D, V. Luonga, VP & Spyratosa, N 1998, 'Updating Intensional Predicates in Deductive Databases', Data & Knowledge Engineering, vol. 26, no. 1, pp. 37-70.

100. Lee, M & Hammer, J 2001, 'Speeding up materialized view selection in data warehouses using a randomized algorithm', International Journal of Cooperative Information Systems, vol. 10, no. 3, pp. 327-343.
101. Liberatore, P 1997, 'The Complexity of Belief Update (Extended in 2003)', proceedings of the Fifteenth International Joint Conference on Artificial Intelligence1, vol. 1, pp. 68-73.
102. Liberatore, P & Schaerf, M 2004, 'The Compactness of Belief Revision and Update Operators', Fundamenta Informaticae, vol. 62, no. 3, pp. 377-393.
103. Liberatore, P & Schaerf, M 2012, 'On the Complexity of Finding Second-Best Abductive Explanations', CoRR abs/1204.5859.
104. Lobo, J, Minker, J & Rajasekar, A 1992, 'Foundations of Disjunctive Logic Programming', MIT Press, Cambridge.
105. Lobo, J & Trajcevski, G 1997, 'Minimal and Consistent Evolution of Knowledge Bases', Journal of Applied Non-Classical Logics, vol:7, no. 1, pp. 117-146.
106. Li, X 2010, 'Materialized View Selection: A Survey', IGI book chapter, View Management Techniques and Their Application to Data Stream Management', DOI: 10.4018/978-1-60566-816-1.ch005.
107. Lu, JJ, Moerkotte, G, Schue, J & Subrahmanian, VS 1995, 'Efficient maintenance of materialized mediated views', Proceedings of the 1995 ACM SIGMOD international conference on Management of data, vol.3, no.2, pp. 340-351.
108. Lu, W 1999, 'View Updates in Disjunctive Deductive Databases Based on SLD-Resolution', Proceedings of the 6th International Workshop on Knowledge Representation meets Databases, ed. Franconi, E, öping, Sweden, vol. 21, pp. 31-35.
109. Maabout, S 1998, 'Maintaining and Restoring Database Consistency with Update Rules', Workshop DYNAMICS, Joint International Conference and Symposium on Logic Programming.
110. Makinson, D 1997, 'Screened Revision', Theoria, vol. 63, pp.14-23.
111. Mami, I, Coletta, R & Bellahseneet, Z 2011, 'Modeling view selection as a constraint satisfaction problem', Database and Expert Systems Applications Lecture Notes in Computer Science, ed. Hameurlain, A, Toulouse, France, vol. 6861, pp. 396-410.
112. Mami, I & Bellahsene, Z 2012, 'A survey of view selection methods]', ACM SIGMOD Record, vol. 41, no. 1, pp. 20-29.
113. Martinenghi, D & Christiansen, H 2005, 'Efficient Integrity Checking for Databases with Recursive Views', Advances in Databases and Information Systems Lecture Notes in Computer Science, ed. Eder, J, Tallinn, Estonia, vol. 3631, pp. 109-124.
114. Mayol, E & Teniente, E 1993, 'Incorporating Modification Requests in Updating Consistent Knowledge Bases', Fourth Int. Works. on the Deductive Approach to Information Systems and Databases, pp. 275-300.
115. Mayol, E & Teniente, E 1999, 'A Survey of Current Methods for Integrity Constraint Maintenance and View Updating', Advances in Conceptual Modeling Lecture Notes in Computer Science, ed. Chen, P, Paris, France, vol. 1727, pp. 62-73.
116. Meyden, R 1998, 'Logical Approaches to Incomplete Information: A Survey', Logics for Databases and Information Systems, pp. 307-356.
117. Minker, J 1996, 'Logic and Databases: A 20 Year Retrospective', Logic in Databases, Lecture Notes in Computer Science, ed. Pedreschi, D, San Miniato, Italy, vol. 1154, pp. 3-57.
118. Moerkotte, G & Lockemann, PC 1991, 'Reactive Consistency Control in Deductive Databases', ACM Transactions on Database Systems, vol. 16 no. 4, pp. 670-702.

119. Mota-Herranz, L, Celma-GimĺȩneZ, M & Decker, H 2000, 'Transaction Trees for Knowledge Revision', Flexible Query Answering Systems Advances in Soft Computing, ed. Prof. Larsen, HL, vol. 7, pp. 182-191.
120. Nayak, A, Chen, Y & Lin, F 2006, 'Forgetting and Knowledge Update', AI 2006: Advances in Artificial Intelligence Lecture Notes in Computer Science, ed. Sattar, A, Hobart, Australia, vol. 4304, pp. 131-140.
121. Nayak, A 2011, 'Is Revision a Special Kind of Update?', AI 2011: Advances in Artificial Intelligence Lecture Notes in Computer Science, ed. Wang, D, Perth, Australia, vol. 7106, pp. 432-441.
122. Nebel, B 1998, 'How Hard is it to Revise a Belief Base?', Handbook of Defeasible Reasoning and Uncertainty Management Systems, pp. 77-145.
123. Özçep, Ö, L 2012, 'Knowledge-Base Revision Using Implications as Hypotheses', KI 2012: Advances in Artificial Intelligence Lecture Notes in Computer Science, ed. Glimm, B, Saarbrücken, Germany, vol. 7526, pp. 217-228.
124. Parikh, R 1999, 'Beliefs, belief revision, and splitting languages', Journal of Logic, language, and Computation, vol. 2, pp. 266-278.
125. Pagnucco, M 1996, 'The Role of Abductive Reasoning within the Process of Belief Revision', PhD Thesis, University of Sydney.
126. Papini, O 2000, 'Knowledge-base revision', The Knowledge Engineering Review, vol. 15, no. 4, pp. 339-370.
127. Pinto, AM & Pereira, LM 2011, 'Each normal logic program has a 2-valued Minimal Hypotheses semantics', 19th International Conference on Applications of Declarative Programming and Knowledge Management, pp. 123-134.
128. Qi, G & Yang, F 2008, 'A Survey of Revision Approaches in Description Logics', Description Logics, Proceedings of the 2nd International Conference on Web Reasoning and Rule Systems, pp. 74-88.
129. Rodrigues, o & Benevidas M 1994, 'Belief Revision in Pseudo-Definite Sets', In Proceeding of the 11th Brazilian Symposium on AI.
130. Rott, H 2001, 'Change, choice and inference: a study of belief revision and nonmonotonic reasoning', Oxford logic guides. Clarendon, Oxford.
131. Sadri. F & Toni, F 2005, 'Interleaving belief revision and reasoning: preliminary report', In Proceedings of Convegno Italiano di Logica Computazionale.
132. Saha, D & Ramakrishnan, CR 2003, 'Incremental Evaluation of Tabled Logic Programs', Logic Programming Lecture Notes in Computer Science Volume, vol. 2916, pp. 392-406.
133. Sakama, C 2011, 'Dishonest Reasoning by Abduction', Proceedings of the Twenty-Second international joint conference on Artificial Intelligence, vol. 2, pp. 1063-1064.
134. Sakama, C & Inoue, K 2009, 'Equivalence issues in abduction and induction', Journal of Applied Logic, vol.7, no. 3, pp. 318-328.
135. Sakama, C & Inoue, K 2003, 'An abductive framework for computing Horn knowledge base updates', Journal Theory and Practice of Logic Programming, vol.3, no. 6, pp. 671-713.
136. Sakama, C & Inoue, K 1999, 'Updating Extended Logic Programs through Abduction', Logic Programming and Nonmonotonic Reasoning Lecture Notes in Computer Science, ed. Gelfond, M, El Paso, Texas, USA, vol. 1730, pp. 147-161.
137. Salman, M, Rehmanet, NU & Shahid, M 2012, 'Database Integrity Mechanism between OLTP and Offline Data', Proceedings of the 4th Asian conference on Intelligent Information and Database Systems, vol. 2, pp. 371-380.

138. Schewe, KD 1996, 'Tailoring Consistent Specializations as a Natural Approach to Consistency Enforcement', 6th Int. Workshop on Foundations of Models and Languages for Data and Objects: Integrity in Databases.
139. Schewe, KD 1998, 'Consistency Enforcement in Entity-Relationship and Object Oriented Models', Data & Knowledge Eng, vol. 28, no. 1, pp. 121-140.
140. Schewe, KD 2000, 'Controlled Automation of Consistency Enforcement', Proceedings of the 15th IEEE international conference on Automated software engineering, pp. 265-268.
141. Schulte, O 1999, 'Minimal Belief Change and Pareto-Optimality', Australian Joint Conference on Artificial Intelligence, pp. 144-155.
142. Segerberg, K 1998, 'Irrevocable Belief Revision in Dynamic Doxastic Logic', Notre Dame Journal of Formal Logic, vol. 39, no.3, pp. 287-306.
143. Siebes, A, Tsur,S, Ullman, JD, Vieille, L & Zaniolo, C 1996, 'Deductive Databases: Challenges, Opportunities and Future Directions (Panel Discussion)', 96 Proceedings of the International Workshop on Logic in Databases, pp. 225-229.
144. Staudt, M, & and Jarke, M 1996, 'Incremental maintenance of externally materialized views', 22th International Conference on Very Large Data Bases, pp. 75-86.
145. Teniente, E & Olive, A 1995, 'Updating Knowledge Bases while Maintaining their Consistency', The Very Large Data Bases Journal, vol. 4, no. 2, pp. 193-241.
146. Urpí, T & Olivé, A 1992, 'A method for change computation in deductive databases', Proceedings of the 18th International Conference on Very Large Data Bases, vol. 225-277.
147. Walliser, B, Zwirnet, D & Zwirn, H 2005, 'Abductive Logics in a Belief Revision Framework', Journal of Logic, Language and Information, vol:14, no.1, pp. 87-117.
148. Wrobel, S 1995, 'First order Theory Refinement', IOS Frontier in AI and Application Series, Advances in ILP, ed. Raedt, LD, IOS Press, Amsterdam.
149. Wu, M, Zhang, D & Zhang, M 2011, 'Language Splitting and Relevance-Based Belief Change in Horn Logic', Twenty-Fifth AAAI Conference on Artificial Intelligence, pp. 268-273.
150. Wüthrich, B 1993, 'On Updates and Inconsistency Repairing in Knowledge Bases', IEEE 9th International Conference on Data Engineering, vol. 608-615.
151. Yang, J, Karlapalem, K & Liet, Q 1997, 'Algorithms for materialized view design in data warehousing environment', Proceedings of the 23rd International Conference on Very Large Data Bases, vol. 136-145.
152. Zanuttini, B 2003, 'New Polynomial Classes for Logic-Based Abduction', Journal of Artificial Intelligence Research, vol. 19, pp. 1-10.
153. Zhang, C & Yang, Y 1999. 'Genetic algorithm for materialized view selection in data warehouse environments', DataWarehousing and Knowledge Discovery, Lecture Notes in Computer Science, ed. Mohani, M, Florence, Italy, vol. 1676, pp. 116-125.
154. Zhou, J, Larson, PA, Goldstein, J & Ding, L (2007), 'Dynamic materialized views', IEEE 23rd International Conference on Data Enginering, pp. 526-535.